

Estimation of Distribution Algorithms based on Copula Functions

by

Rogelio Salinas Gutiérrez

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Science
in
Computer Science

Centro de Investigación en Matemáticas
Guanajuato, México

Public Defense on Tuesday, August 09, 2011

This thesis has been conducted under the supervision of:

Dr. Arturo Hernández Aguirre, thesis advisor

Dr. Enrique Raúl Villa Diharce, thesis co-advisor

The doctoral thesis committee:

Dr. Elva Díaz Díaz Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Aguascalientes

Dr. Jean Bernard Hayet Centro de Investigación en Matemáticas

Dr. Rogelio Ramos Quiroga Centro de Investigación en Matemáticas

Dr. Salvador Ruiz Correa Centro de Investigación en Matemáticas

Dr. Enrique Raúl Villa Diharce Centro de Investigación en Matemáticas

This document was typeset by the author using L^AT_EX.

© Copyright 2011 by Rogelio Salinas Gutiérrez

All rights reserved. No part of this publication may be reproduced, transmitted or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without the prior permission of the author.

The dissertation entitled “Estimation of Distribution Algorithms based on Copula Functions”, conducted by Rogelio Salinas Gutiérrez in partial fulfillment of the requirements for the degree of Doctor of Science in Computer Science has been examined, approved and accepted as to style and content by:

The Thesis Committee Members

Dr. Rogelio Ramos Quiroga, Chairperson

Dr. Salvador Ruiz Correa, Secretary

Dr. Elva Díaz Díaz, Thesis Reader

Dr. Jean Bernard Hayet, Thesis Reader

Dr. Enrique Raúl Villa Diharce, Thesis Coadvisor

The Academic Coordination

Dr. Arturo Hernández Aguirre, Academic Coordinator and Thesis Advisor

Centro de Investigación en Matemáticas
2011

*To my wife Leticia,
my daughter Sofía Carolina and
my son Rogelio Emiliano.*

Abstract

Estimation of Distribution Algorithms based on Copula Functions

by

Rogelio Salinas Gutiérrez

Doctor of Science in Computer Science

Centro de Investigación en Matemáticas

Guanajuato, México, 2011

Dr. Arturo Hernández Aguirre, advisor

Dr. Enrique Raúl Villa Diharce, co-advisor

An important paradigm for solving continuous optimization problems has been the use of the multivariate normal distribution in Estimation of Distribution Algorithms (EDAs). However, as a consequence, linear dependencies among variables in the selected population along with marginal and conditional normal distributions must be assumed. These conditions could not be realistic for some optimization problems.

This research work presents some novel proposals for modeling the dependence structure of the selected individuals in continuous EDAs. The followed research approach has been to model the most important dependencies in the selected population and to estimate their associated parameters in the corresponding multivariate distribution. Contributions of this doctoral dissertation are, among others, the use of copula entropies for building the graphical model, the incorporation of a procedure for selecting the most adequate copula function, and the generalization of some well known EDAs.

Resumen

Estimation of Distribution Algorithms based on Copula Functions

por

Rogelio Salinas Gutiérrez

Doctor en Ciencias con orientación en Ciencias de la Computación

Centro de Investigación en Matemáticas

Guanajuato, México, 2011

Dr. Arturo Hernández Aguirre, asesor

Dr. Enrique Raúl Villa Diharce, co-asesor

Un paradigma importante para resolver problemas de optimización en dominio continuo ha sido el uso de la distribución normal multivariada en los Algoritmos de Estimación de Distribución (EDAs). Sin embargo, como consecuencia, debe suponerse que las dependencias entre variables de la población seleccionada son del tipo lineal y que las distribuciones marginales y condicionales son también normales. Estas condiciones podrían no ser realistas para algunos problemas de optimización.

Este trabajo de investigación presenta algunas propuestas nuevas para modelar la estructura de dependencia entre los individuos seleccionados en EDAs continuos. El enfoque seguido para esta investigación ha sido modelar las dependencias más importantes en la población seleccionada y estimar sus parámetros asociados en la correspondiente distribución multivariada. Algunas de las contribuciones de esta tesis son el uso de entropías de cópula para construir el modelo gráfico, la incorporación de un procedimiento para seleccionar la función de cópula más adecuada y la generalización de algunos EDAs bien conocidos.

Contents

Signature Page	i
Dedication	iii
Abstract	v
Resumen	vii
Contents	ix
List of Figures	xiii
List of Tables	xv
List of Algorithms	xvii
List of Notations	xix
List of Abbreviations	xxi
1 Introduction	1
1.1 Motivation	2
1.2 Objectives	3
1.3 Background and related works	3
1.4 Methodology	5
1.5 Structure of the thesis	5
2 Estimation of Distribution Algorithms	7
2.1 Introduction	7
2.2 Univariate EDAs	8
2.2.1 UMDA	8
2.2.2 PBIL	9
2.2.3 Other univariate EDAs	9
2.3 Bivariate EDAs	10
2.3.1 MIMIC	10

2.3.2	Dependence trees	11
2.3.3	BMDA	11
2.4	Multivariate EDAs	12
2.4.1	EcGA	12
2.4.2	Bayesian factorizations	13
2.4.3	Multivariate Gaussian distributions	14
2.5	Summary	14
3	Copula Functions	17
3.1	Introduction	17
3.2	Bivariate copula functions	19
3.2.1	Fréchet-Hoeffding bounds and the product copula	20
3.2.2	Some bivariate copulas	21
3.2.3	Estimation of the parameter	24
3.2.4	Sampling from a copula function	24
3.3	Multivariate copulas	25
3.3.1	The Gaussian copula	26
3.3.2	Archimedean copulas	29
3.4	Summary	31
4	EDAs and Copula Functions	33
4.1	Introduction	33
4.2	Copula functions and graphical models	34
4.2.1	The chain graphical model	35
4.2.2	The tree graphical model	36
4.2.3	Copula entropy and mutual information	37
4.2.4	Sampling from Bayesian networks	40
4.3	Our proposed EDAs	41
4.3.1	Incorporating copula functions	41
4.3.2	Incorporating a copula selection procedure	46
4.4	Summary	54
5	The D-vine EDA	55
5.1	Regular vines	55
5.2	Copulas and information theory measures	59
5.3	The proposed EDA	63
5.3.1	Incorporating a new graphical model	67
5.4	Summary	70
6	Conclusions	71
	Bibliography	75
	Appendices	

A Selected copula functions	85
A.1 Ali-Mikhail-Haq	85
A.2 Clayton	86
A.3 Farlie-Gumbel-Morgenstern	86
A.4 Frank	87
A.5 Gaussian	87
A.6 Gumbel	88
B Benchmark functions	89
B.1 Ackley	89
B.2 Cigar	89
B.3 Cigar Tablet	90
B.4 Ellipsoid	90
B.5 Griewangk	90
B.6 Rastrigin	91
B.7 Rosenbrock	91
B.8 Schwefel 1.2 (Quadric)	91
B.9 Sphere Model	92
B.10 Trid	92
B.11 Two Axes	92
B.12 Zakharov	93
Acknowledgements	95
About the author	97

List of Figures

2.1	A chain graphical model	11
2.2	A tree graphical model	11
2.3	A forest of trees model	12
3.1	Graphs of Fréchet-Hoeffding bounds and the product copula . . .	21
3.2	Graphs of FGM, Frank and Gaussian copulas	23
3.3	Samples from a Frank copula	26
3.4	Samples from a Gaussian copula	28
4.1	A chain graphical model and a tree graphical model	34
4.2	A chain graphical model and a tree graphical model with copula functions	39
4.3	Success rate for separable functions	52
4.4	Success rate for non-separable functions	53
5.1	A pair copula decomposition for a trivariate density function . .	56
5.2	Example of a four-dimensional C-vine.	58
5.3	Example of a four-dimensional D-vine.	59

List of Figures

List of Tables

3.1	Bivariate Archimedean copulas.	32
4.1	Descriptive fitness results for all test functions.	44
4.2	Descriptive results for function evaluations in all test functions. .	45
4.3	Results for the difference between fitness means in each problem. A 95% interval confidence and a p-value are obtained through a Bootstrap technique.	46
4.4	Descriptive results of the fitness for separable functions.	49
4.5	Descriptive results of the fitness for non-separable functions. . . .	50
5.1	Amount of information given for each tree in a C-vine.	64
5.2	Amount of information given for each tree in a D-vine.	65
5.3	Descriptive fitness results for all test functions.	68
5.4	Descriptive function evaluations for all test functions.	69
5.5	Results for the difference between fitness means in each problem	70

List of Tables

List of Algorithms

1	Pseudocode for EDAs	8
2	Pseudocode for generating samples from a bivariate copula function	25
3	Pseudocode for estimating Gaussian copula parameters	27
4	Pseudocode for generating data from a Gaussian copula	27
5	Greedy algorithm to pick a permutation α in a chain model . . .	40
6	Monte Carlo method for estimating the copula entropy	40
7	Pseudocode of the PLS	41
8	Greedy algorithm to pick a permutation α	42
9	Pseudocode for estimating the model and generating a new population	43
10	Greedy algorithm to pick a permutation γ in a D-vine	66
11	Pseudocode for estimating the model and generating a new population	67

List of Algorithms

List of Notations

d	dimension space
n	size of selected individuals
$p(x_i)$	univariate marginal probability of discrete variable X_i , $p(X_i = x_i)$
$p(x_i x_j)$	conditional probability of X_i given $X_j = x_j$, $p(X_i = x_i X_j = x_j)$
$p(\mathbf{x})$	joint probability of discrete random vector \mathbf{X} , $p(\mathbf{X} = \mathbf{x})$
$f(x_i)$	univariate marginal density function of continuous variable X_i , $f(X_i = x_i)$
$f(x_i x_j)$	conditional density function of X_i given $X_j = x_j$, $f(X_i = x_i X_j = x_j)$
$f(\mathbf{x})$	joint density function of continuous random vector \mathbf{X} , $f(\mathbf{X} = \mathbf{x})$
G	graph $G = (V, E)$
V	set of vertices in a graph
E	set of edges in a graph

List of Notations

List of Abbreviations

AMaLGaM-IDEA	Adapted Maximum-Likelihood Gaussian Model IDEA
BMDA	Bivariate Marginal Distribution Algorithm
BIC	Bayesian Information Criterion
BOA	Bayesian Optimization Algorithm
CEC	Congress on Evolutionary Computation
cGA	compact Genetic Algorithm
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
COMIT	Combining Optimizers with Mutual Information Trees
DEUM	Distribution Estimation Using Markov Random Field
D-vine EDA	EDA based on Regular Vines
EA	Evolutionary Algorithm
EBNA	Estimation of Bayesian Network Algorithm
EBNA _{BIC}	EBNA using BIC metric
EBNA _{K2+pen}	EBNA using K^2 algorithm with a penalising factor
EBNA _{PC}	EBNA using the PC algorithm
EC	Evolutionary Computation
ECGA	Extended cGA
EDA	Estimation of Distribution Algorithm
EGNA	Estimation of Gaussian Network Algorithm
EMNA	Estimation of Multivariate Normal Algorithm
GA	Genetic Algorithm
GECCO	Genetic and Evolutionary Computation Conference
IDEA	Iterated Density-Estimation Evolutionary Algorithm
MBOA	Mixed Bayesian Optimization Algorithm
MIMIC	Mutual Information Maximization Input Clustering
MIMIC _c ^G	MIMIC for Gaussian models
PADA	Polytree Approximation of Distribution Algorithm
PBIL	Population-Based Incremental Learning
PBIL _c	Continuous PBIL

List of Abbreviations

PLS	Probabilistic Logic Sampling
PMBGA	Probabilistic Model Building Genetic Algorithm
PPSN	Parallel Problem Solving from Nature
rBOA	real-coded Bayesian Optimization Algorithm
RELEDA	Reinforcement Learning Estimation of Distribution Algorithm
UMDA	Univariate Marginal Distribution Algorithm
UMDA _c	UMDA for continuous domains
UMDA _c ^G	UMDA for Gaussian models

Chapter 1

Introduction

Nowadays, the optimization methods have been recognized as important tools for finding optimal solutions in several fields, such as Computer Science, Statistics, Artificial Intelligence, Operations Research, among others. In general, an optimization problem can be modeled by a mathematical function. The feasible set of solutions for an optimization model is represented by a set of decision variables. The quality of the decision variables is given by a function, commonly named *objective function*. The interest in optimization consists of finding the best possible solution in a reasonable period of time.

The optimization problems have been studied and solved with different proposals. Some of these proposals, named *metaheuristics*, have been designed for solving hard optimization problems. Although this kind of algorithms does not guarantee global optimal, in practice, they usually find good solutions in a reasonable period of time. The *Evolutionary Computation* (EC) is a subfield of artificial intelligence that consists of metaheuristic techniques for solving optimization problems. These metaheuristics use principles of Darwin's theory and they are also known as *Evolutionary Algorithms* (EAs). Each iteration in an EA involves a competitive *selection* that chooses the best solutions. The solutions with highest fitness are *crossed over* for creating new solutions. Some individuals of the new population can be mutated in order to preserve diversity in the solutions. In this way, the genetic operators cross over and mutation are used for giving *variation* to the set of solutions.

Estimation of Distribution Algorithms (EDAs) are a new class of evolutionary optimization techniques that employ probabilistic models as a representation of the relationships between variables in the population. This recent paradigm in EC does not use genetic operators such as crossover and mutation. The goal in EDAs is to model the dependencies in the best individuals and transfer them into the next population. EDAs generate the new population by sampling from the probabilistic model of promissory individuals. These evolutionary optimization techniques have used several probabilistic models. For this reason, there are a number of EDAs for discrete and continuous domains. Some of these probabilistic models are based on Bayesian and Markov networks. Other EDAs

have used Gaussian assumptions, such as Gaussian kernels, Gaussian mixture models and the multivariate Gaussian distribution.

Although there have been many applications and studies related to discrete EDAs, it is interesting to design new EDAs for continuous domains. More specifically, it is important to *design multivariate distributions that represent satisfactorily dependencies among continuous variables and that are relatively easy to estimate and to sample.*

On the other hand, in different areas such as finance, climate, oceanography, hydrology, geodesy, and reliability researchers have used probabilistic models that separate marginal distributions and the dependence structure. This has let more flexibility for modeling multivariate data, without necessity of restricting the marginal distributions. The way in which this can be done is by means of the *copula functions*. In the last years, copula functions have become an important option for modeling multivariate data.

This thesis has been developed in the research area of EC, with the aim of implementing copula functions in continuous EDAs. In the following two sections we present the motivation as well as the objectives of this work. We finish this chapter by presenting the related work and the methodology.

1.1 Motivation

One motivation for this research is the fact that EDAs have the capacity of taking into account *explicitly* dependencies among variables in optimization problems. This characteristic, along with the possibility of transferring dependencies into the next generation of new solutions, have received much attention from the EC community. An important goal of modeling dependencies among variables in an EDA is to learn the structure of the optimization problem (Bosman and Grahl, 2005; Grahl et al., 2006). The learning of the problem structure by means of a probabilistic model can help ensure an efficient optimization behavior.

Although EDAs have been investigated for discrete and continuous domains, the works and contributions for continuous domains are mainly based on the multivariate Gaussian distribution. The assumption of modeling dependencies under this probabilistic model can not be realistic for some optimization problems. This observation gives an opportunity, and a motivation, for proposing new continuous EDAs.

Another motivation is related to the growing use of copula functions for getting flexible multivariate distributions. This is because of the important contributions that copula theory has had in many research and application works, such as finance Cherubini et al. (2004); Trivedi and Zimmer (2007), climate Schölzel and Friederichs (2008), oceanography De-Waal and Van-Gelder (2005), hydrology Genest and Favre (2007), geodesy Bacigál and Komorníková (2006), and reliability Monjardin (2007).

1.2 Objectives

The main goal of this doctoral research is to study the incorporation of copula functions into continuous EDAs. In particular, the objectives are the following:

General objective

- To investigate the copula theory for incorporating copula functions into the modeling of dependencies among variables.

Specific objectives

- To study probabilistic models based on copula functions and to implement them in continuous EDAs.
- To study the methods for estimating copula parameters and the procedures for sampling from copula functions. If required, to propose algorithms for learning copula parameters and sampling from copulas.
- To implement an copula based EDA in a programming language.
- To evaluate the performance of a copula based EDA and compare statistically the results against other EDAs.

1.3 Background and related works

It is known that EDAs models explicitly the dependencies among variables for solving optimization problems. This capacity is not presented in other EAs such as the Genetic Algorithm (GA). Some representative works in this direction are the papers presented by Heinz Mühlenbein (Mühlenbein and Paaß, 1996; Mühlenbein, 1998; Mühlenbein et al., 1999). These studies make an important contribution for considering a new and promissory class of EAs in the EC community.

The research in EDAs has been motivated by their capacity of taking into account the interactions between variables. This source of knowledge has been called *linkage information* and was investigated by different authors for extending simple GAs to process interrelations, i.e., *building blocks* (see Larrañaga, 2002b). Furthermore, it is possible to make theoretical analysis of the evolutive process in EDAs (González, 2005).

EDAs have become a growing field into the EC community. Nowadays the works in EDAs are presented in the three most important conferences of EC: *Genetic and Evolutionary Computation Conference (GECCO)*, *Congress on Evolutionary Computation (CEC)* and *Parallel Problem Solving from Nature (PPSN)*.

According to Höns (2005), the EDA track for the GECCO and CEC appeared in 2005. The publication of papers in several journals and the publication of books such as (Larrañaga and Lozano, 2002; Pelikan et al., 2006), the presentation of works in many other conferences, along with other academic activities like seminars and workshops, give evidence that the research on EDAs is an active research area in EC. For example, some doctoral dissertations have been conducted for proposing and studying probabilistic models in discrete (Pelikan, 2002; Soto-Ortiz, 2003; Santana-Hermida, 2004; Shakya, 2006) and continuous domains (Bosman, 2003). A study about the minimum relative entropy and EDAs is presented in the doctoral work of Höns (2005). By means of Markov chains and dynamical systems, a theoretical analysis for univariate EDAs is done in the doctoral thesis of González (2005). Other doctoral theses apply EDAs to the graph matching problem (image recognition) (Bengoetxea, 2002), the optimization of composite laminates (Grosset, 2004), calibration models for quantitative chemical applications (Mendiburu-Alberro, 2006), and combinatorial problems in graphical models (Romero-Asturiano, 2007). An investigation about the implementation of parallel EDAs is shown in Očenášek (2002).

Nowadays there are several EDAs for optimization problems in discrete and continuous domains. The EDAs can be classified as univariate, bivariate or multivariate according to the complexity of their probabilistic model used to learn the interactions between the variables. The univariate EDAs consider all the variables as independent, for instance, the Univariate Marginal Distribution Algorithm (UMDA) (Mühlenbein, 1998; Larrañaga et al., 1999, 2000a), the Population Based Incremental Learning (PBIL) (Baluja, 1994), and the compact Genetic Algorithm (cGA) (Harik et al., 1998). The bivariate EDAs take into account dependencies between pairs of variables and a few examples are the Bivariate Marginal Distribution Algorithm (BMDA) (Pelikan and Mühlenbein, 1999), Mutual Information Maximizing Input Clustering (MIMIC) (De Bonet et al., 1997; Larrañaga et al., 1999, 2000a), and Dependency-Trees (Baluja and Davies, 1997). Many univariate and bivariate discrete EDAs have been extended to continuous domains by using Gaussian probabilistic models.

For multiple dependencies in discrete domain the EDAs have used probabilistic models such as the Polytrees Approximation of Distribution Algorithm (PADA) (Soto et al., 1999), Estimation of Bayesian Network Algorithm (EBNA) (Etxeberria and Larrañaga, 1999; Larrañaga et al., 2000b) and Bayesian Optimization Algorithm (BOA) (Pelikan et al., 1999).

For real-valued (continuous) multivariate variables the EDAs have used mostly multivariate Gaussian distributions and some examples are the Estimation of Multivariate Normal Algorithm (EMNA) (Larrañaga et al., 2001) and Estimation of Gaussian Network Algorithm (EGNA) (Larrañaga et al., 1999, 2000a). The EDA AMaLGaM (Bosman et al., 2008) and the algorithm CMA-ES (Igel et al., 2006) are also based on the multivariate Gaussian distribution. Both algorithms modify the estimated covariance matrix in order to improve the convergence rate towards the optimum. Currently they are the state of the art in real-valued optimization.

To the best of our knowledge the theses presented by Barba-Moreno (2007)

and Arderí-García (2007) are the first attempts to incorporate a multivariate Gaussian copula function in EDAs. Since then, other related works have been published. These papers present EDAs based on (1) the Gaussian copula function (Wang et al., 2009a; Wang and Zeng, 2010), and (2) Archimedean copula functions with a fixed dependence parameter (Wang et al., 2009b, 2010a,c; Cuesta-Infante et al., 2010; Wang et al., 2010b). Unlike the previous papers that use Archimedean copula functions, the works of Flores de la Fuente (2009) and Gao (2009) present a way of estimating the copula parameter. All these works, with exception of Flores de la Fuente (2009), only use multivariate copula functions to model the dependence structure among decision variables and do not employ graphical models. On the other hand, in (Salinas Gutiérrez et al., 2009; Salinas-Gutiérrez et al., 2009, 2010c, 2011b), we have proposed the use of the maximum likelihood method and copula entropies in order to (1) estimate the copula parameters, and (2) build a graphical model which establishes the most important dependencies between variables. One recent contribution of our research work (Salinas-Gutiérrez et al., 2011b,a), it has been the incorporation of a procedure for selecting the most adequate copula function.

1.4 Methodology

The performance of proposed algorithms will be tested from an empirical perspective. The numerical results will be

Metodolog a El desempe o del algoritmo se basa en su habilidad para encontrar el valor ptimo: aptitud y tasa de xitos. El desempe o del algoritmo es estudiado desde una perspectiva emp rica: un algoritmo se ejecuta varias veces en un conjunto de funciones de prueba. Los datos emp ricos se obtienen de cada corrida: la mejor aptitud, n mero de evaluaciones de funci n (FE). El uso de criterios de paro: convergencia local, soluci no ptima encontrada y n mero m ximo de generaciones o FE. El desempe o de algoritmos se compara por medio de herramientas estad sticas: prueba de hip tesis y t cnicas bootstrap.

The performance of the proposed algorithms is based their ability for finding the optimum in an optimization problem on numerical The ability of an algorithm for finding the optimum will be taken into account In order to evaluate the performance of the proposed algorithms

1.5 Structure of the thesis

The structure of the thesis is the following: chapter 2 presents a basic review of EDAs, chapter 3 provides a brief introduction to copula functions, chapter 4 describes the implementation of copula functions in the chain and tree graphical models and explains the copula selection procedure, chapter 5 presents regular vines as well as some theoretical connections between the information theory and the copula functions. Final remarks, contributions and future work are commented in chapter 6.

Chapter 2

Estimation of Distribution Algorithms

A recent paradigm in Evolutionary Computation (EC) that deals with optimization problems is the use of probabilistic models for searching and generating promissory solutions. Algorithms based on this principle have been called Estimation of Distribution Algorithms (EDAs), Probabilistic Model Building Genetic Algorithms (PMBGAs), and also as Iterated Density Estimation Algorithms (IDEAs). In this chapter, we present a general introduction to this well established class of Evolutionary Algorithms (EAs) along with a brief description of some well known EDAs.

2.1 Introduction

In this dissertation, the class of EAs that employs probabilistic models as a representation of the relationships among variables is identified as EDAs. Similar to Genetic Algorithms (GAs), EDAs are population based. However, this new class of EAs does not use genetic operators such as crossover and mutation for generating new individuals. Instead, in EDAs, the new individuals are sampled from a probability distribution. Therefore, the goal in EDAs is to take into account the dependencies of the best solutions and transfer them into the next population. A pseudocode for EDAs is shown in Algorithm 1. The use of the estimated model in step 4 allows one to explicitly take into account the dependencies between decision variables and their structure. Step 5 shows the possibility of incorporating the dependencies among the variables into the new population, which greatly modifies the performance of an EDA.

The main advantage of using probabilistic distributions in evolutionary algorithms is that the interrelations among the variables of a population are *explicitly* modeled. Thus, according to step 4 of Algorithm 1, the estimation of a probabilistic model is an important procedure in EDAs. Therefore, ever since

Algorithm 1 Pseudocode for EDAs

- 1: Initialize the generation counter $t \leftarrow 0$
 Generate the initial population \mathcal{P}_0 with N individuals at random.
 - 2: Evaluate population \mathcal{P}_t using the cost function.
 - 3: Select a subset \mathcal{S}_t from \mathcal{P}_t according to the selection method.
 - 4: Estimate a probabilistic model \mathcal{M}_t from \mathcal{S}_t .
 - 5: Generate the new population \mathcal{P}_{t+1} by sampling from the model \mathcal{M}_t
 Assign $t \leftarrow t + 1$.
 - 6: If stopping criteria are not reached go to step 2.
-

EDAs were introduced in the field of EC, many researchers have been interested in proposing and enhancing new probabilistic models.

A number of EDAs have been proposed for optimization problems in discrete and continuous domains. EDAs can be classified according to the complexity of their probabilistic model used to learn the interactions between the variables. The following sections present some of the most representative EDAs for discrete and continuous search spaces.

2.2 Univariate EDAs

The univariate EDAs assume independence among all the variables. The independence assumption reduces the complexity of a joint probabilistic distribution. Thus, these algorithms do not take into account interactions between variables. For these EDAs, the joint probabilistic distribution can be estimated as the product of marginal distributions.

2.2.1 UMDA

The Univariate Marginal Distribution Algorithm (UMDA) was introduced by Mühlenbein and Paaß (1996) for discrete domains and it is considered as one of the early works in EDAs. The probabilistic model used by UMDA is the univariate factorization for binary random variables:

$$P(X_1, \dots, X_d) = \prod_{i=1}^d P(X_i) \tag{2.1}$$

where X_i is a binary random variable, $X_i \in \{0, 1\}$. In order to estimate the probabilistic model (2.1), the univariate marginal frequencies are calculated.

For continuous domains, the UMDA was adapted by Larrañaga et al. (1999, 2000a). This adaptation is known as Univariate Marginal Distribution Algorithm for continuous domains (UMDA_c). Unlike the UMDA, the probabilistic model for UMDA_c identifies the univariate structure in the sense that marginal densities are selected via hypothesis tests. Once the densities have been identified, the estimation of the parameters is performed by their maximum likelihood

estimates.

In the particular case that all univariate densities are Gaussian distributions, the UMDA_c is called Univariate Marginal Distribution Algorithm for Gaussian models (UMDA_c^G).

For both domains, binary and real-valued random variables, the parameters of the marginal distributions are estimated by the maximum likelihood method.

2.2.2 PBIL

The Population-Based Incremental Learning (PBIL) algorithm was proposed by Baluja (1994); Baluja and Caruana (1995) for solving optimization problems in discrete domains. This algorithm uses a probability vector for representing the population of individuals. At each iteration, individuals are generated by sampling the probability vector and some of the best individuals are selected. The selected set of individuals is used to update the probability vector by means of the following rule:

$$p_i = (1 - \alpha)p_i + \alpha p(x_i), \quad \text{for } i = 1, \dots, d \quad (2.2)$$

where p_i represents the probability of 1 for the i^{th} binary random variable in the solution, $p(x_i)$ is the estimated univariate marginal probability for each variable x_i , and α is a learning rate parameter between 0 and 1. The use of an update rule for the probability vector gives PBIL a memory of previous good solutions.

Some extensions of PBIL to continuous search spaces are proposed in (Rudlof and Köppen, 1996; Servet et al., 1997; Sebag and Ducoulombier, 1998). The algorithm presented in (Sebag and Ducoulombier, 1998), named PBIL_c , uses a Gaussian model for the distribution of the population. At each generation, the mean vector $\boldsymbol{\mu}$ is updated from a linear combination of the best two and the worst individuals in the current population,

$$\boldsymbol{\mu}^{(t+1)} = (1 - \alpha) \cdot \boldsymbol{\mu}^{(t)} + \alpha \cdot (\mathbf{x}^{\text{best},1} + \mathbf{x}^{\text{best},2} - \mathbf{x}^{\text{worst}}) \quad (2.3)$$

For the adaptation of the vector of variances, $\boldsymbol{\sigma}$, four heuristics are proposed. One of these proposals calculates the sample variance of the K best current individuals. The other heuristics are: (1) to use a constant value; (2) to use a self-adaptive $(1, \lambda)$ evolution strategy; (3) to calculate a linear combination between the previous variance and the current sample variance.

2.2.3 Other univariate EDAs

cGA

The compact Genetic Algorithm (cGA) by Harik, Lobo, and Goldberg (1998) was proposed for binary representations. Similar to PBIL, the cGA also uses a probability vector. However, at each generation, the cGA updates the probability vector by using the best of only two individuals.

RELEDA

The Reinforcement Learning Estimation of Distribution Algorithm (RELEDA) by Paul and Iba (2003) uses reinforcement learning and marginal probability distribution of selected individuals in order to generate a new population of solutions. The RELEDA was proposed for optimization in discrete domains.

DEUM

The Distribution Estimation Using Markov Random Field (DEUM) by Shakya (2006) proposes the use of the fitness function to estimate the parameters of the distribution. This property distinguishes DEUM from other EDAs. Several variants of DEUM are presented in (Shakya, 2006) by using different techniques to sample Markov Random Fields.

2.3 Bivariate EDAs

Algorithms in this second category consider only pairwise interactions among variables. Therefore, unlike univariate EDAs, the construction of a structure is necessary in order to represent conditional distributions in the probabilistic model. This class of algorithms has a good performance in problems where pairwise interaction among variable exists.

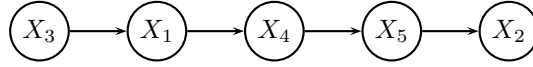
2.3.1 MIMIC

The Mutual Information Maximizing Input Clustering (MIMIC) by De Bonet, Isbell, and Viola (1997) proposes the use of a chain model for representing bivariate dependencies among variables. The probabilistic model for discrete domains is the following:

$$P(\mathbf{x}) = P(x_{\alpha_1}) P(x_{\alpha_2}|x_{\alpha_1}) \cdots P(x_{\alpha_d}|x_{\alpha_{(d-1)}}) , \quad (2.4)$$

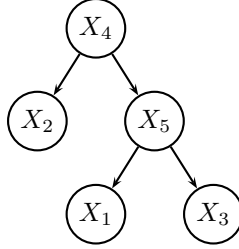
where $\mathbf{x} = (x_1, \dots, x_d)$ is a vector of variables and $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d)$ is a permutation of the integers between 1 and d . At each generation, the MIMIC estimates the factorization (2.4) by searching for the best permutation between the variables. A greedy algorithm is used to find a sub-optimal permutation $\boldsymbol{\alpha}$. The greedy algorithm is based on the minimization of the Kullback-Leibler between the full multivariate probability distribution and the proposed chain model (2.4). Figure 2.1 shows an example of a chain graphical model.

The extension of the MIMIC algorithm to the continuous domain was proposed by Larrañaga, Etxeberria, Lozano, and Peña (1999, 2000a). This adaptation, named MIMIC_c^G, assumes that the underlying probability model for every pair of variables is represented by a bivariate Gaussian distribution. The joint density function is factorized by a chain structure, fitting the model as close as possible to the empirical data by using one univariate marginal density and $d - 1$ pairwise conditional density functions.



$$P(x_1, x_2, x_3, x_4, x_5) = P(x_3) P(x_1|x_3) P(x_4|x_1) P(x_5|x_4) P(x_2|x_5)$$

Figure 2.1: A chain graphical model over five variables.



$$P(x_1, x_2, x_3, x_4, x_5) = P(x_4) P(x_2|x_4) P(x_5|x_4) P(x_1|x_5) P(x_3|x_5)$$

Figure 2.2: A tree graphical model over five variables.

2.3.2 Dependence trees

The Combining Optimizers with Mutual Information Trees by Baluja and Davies (1997) uses a tree model for representing the pairwise interaction among discrete variables. The COMIT algorithm employs the following probabilistic model:

$$P(\mathbf{x}) = P(x_{\beta_1}) P(x_{\beta_2}|x_{\beta_{p(2)}}) \cdots P(x_{\beta_d}|x_{\beta_{p(d)}}) \quad , \quad (2.5)$$

where $\beta = (\beta_1, \dots, \beta_d)$ is a permutation of integers $1, \dots, d$, and $p(i)$ maps numbers $1 < i \leq d$ to integers $1 \leq p(i) < i$. It can be noticed that each variable in the factorization (2.5) is conditioned upon at most one parent. A graphical example of this probabilistic model can be seen in Figure 2.2.

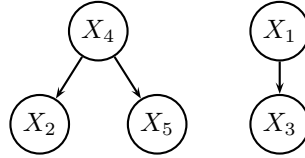
In COMIT, the factorization (2.5) is estimated by means of an algorithm by Chow and Liu (1968) that is guaranteed to get the maximum likelihood tree factorization.

2.3.3 BMDA

The Bivariate Marginal Distribution Algorithm (BMDA) by Pelikan and Mühlenbein (1999) uses a forest of trees as probabilistic model. The factorization is given by

$$P(\mathbf{x}) = \prod_{r \in R} P(x_r) \prod_{i \in V \setminus R} P(x_i|x_{j(i)}) \quad , \quad (2.6)$$

where V denotes the set of d binary variables, $R \subseteq V$ denotes the set of



$$P(x_1, x_2, x_3, x_4, x_5) = P(x_4) P(x_2|x_4) P(x_5|x_4) P(x_1) P(x_3|x_1)$$

Figure 2.3: A forest of trees model over five variables.

root variables, and $X_{j(i)}$ returns the variable connected to the variable X_i and added before X_i . As can be noticed, the BMEDA can have more than one root variable. Figure 2.3 shows a graphical example of the factorization (2.6) for two trees over five variables.

In the BMEDA, the factorization is built by means of Pearson's χ^2 test on the dependencies between pairs of variables.

2.4 Multivariate EDAs

Several EDAs have been proposed in the literature considering interaction between variables of order greater than two to factorize the probability distribution. The probabilistic model of this class of algorithms is more complex than the one used by univariate and bivariate EDAs. For this reason, an advantage is that dependencies between variables can be expressed properly, whereas a drawback is that the probability model required for some problems can be excessively complex and, sometimes, computationally infeasible to search through all possible models.

2.4.1 EcGA

The Extended compact Genetic Algorithm (ECGA) by Harik (1999) is a multivariate extension of the cGA for binary variables. The ECGA uses a greedy algorithm for detecting groups of variables. For each group, a multivariate distribution is estimated. Thus, the probabilistic model used in ECGA is given by the following factorization

$$P(\mathbf{x}) = \prod_{c \in m} P(\mathbf{x}_c) \quad , \quad (2.7)$$

where m is the set of disjoint subsets and $P(\mathbf{x}_c)$ is the multivariate marginal distribution of the group of variables \mathbf{x}_c in the subset c . In (Harik, 1999), the factorization (2.7) is named Marginal Product Models (MPM).

The complexity of the MPM is taken into account for detecting non overlapping groups of variables. This complexity metric is based on (1) the sum of

the entropies of the marginal distributions, and (2) the Minimum Description Length (MDL).

2.4.2 Bayesian factorizations

Bayesian networks have been employed as probabilistic models in EDAs. A Bayesian network consists of a set of variables and a set of directed edges between variables. In general, a Bayesian network specifies a unique joint probability distribution $P(\mathbf{X})$ given by the product of all conditional probability distributions:

$$P(\mathbf{X}) = \prod_{i=1}^d P(X_i | \text{pa}(X_i)) \quad , \quad (2.8)$$

where $\text{pa}(X_i)$ are the parents of X_i . Some graphical examples of Bayesian networks are shown in Figures 2.1, 2.2 and 2.3. An important property for Bayesian networks is that directed cycles are not allowed in its structure.

The following multivariate EDAs employ probabilistic models based on Bayesian networks. Unlike the bivariate EDAs described in the last section, these algorithms are able to model dependencies among two or more variables.

EBNA

The Estimation of Bayesian Network Algorithms (EBNA) was proposed for discrete domains in the work of Etxeberria and Larrañaga (1999) and Larrañaga, Etxeberria, Lozano, and Peña (2000b). At each generation, the selected individuals are taken into account for learning the structure of the Bayesian network. Depending on the method used for learning the structure, the EBNA presents three different variants: the EBNA_{BIC} , the EBNA_{K2+pen} , and the EBNA_{PC} . These algorithms are respectively based on (1) the Bayesian Information Criterion (BIC), (2) the $K2$ algorithm with a penalising factor, and (3) the PC algorithm.

LFDA

Similar to EBNA_{BIC} , the Learning Factorized Distribution Algorithm (LFDA) by Mühlenbein and Mahnig (1999) also uses the BIC metric for building a Bayesian factorization. However, the main difference is that in the LFDA the complexity of the structure is controlled by the BIC metric and a restriction on the maximum number of parents that each variable can have in the Bayesian network.

BOA

The Bayesian Optimization Algorithm (BOA) by Pelikan, Goldberg, and Cantú-Paz (1999) uses the Bayesian Dirichlet metric for measure the goodness of each structure. At each generation, the BOA starts with an empty structure. In

order to reduce the cardinality of the search space, an explicit limit on the number of parents per variable is used.

PADA

The Polytree Approximation of Distribution Algorithm (PADA) by Soto, Ochoa, Acid, and de Campos (1999) uses a Bayesian network with a polytree structure for binary variables. A polytree is a directed graph having a tree for its underlying undirected graph, i.e., there is no more than one undirected path connecting every pair of variables. For constructing the polytree, conditional dependencies are detected. A detailed study of PADA is done in (Soto-Ortiz, 2003).

2.4.3 Multivariate Gaussian distributions

For real-valued variables, the multivariate Gaussian distribution has been widely used as probabilist model. Besides the Gaussian density can be used as a joint distribution, it is a very flexible model for being employed in kernels, mixture distributions and Bayesian networks.

EMNA

The Estimation on Multivariate Normal Algorithm (EMNA) by Larrañaga, Lozano, and Bengoetxea (2001) uses the multivariate Gaussian density for modeling the dependencies among the variables of the selected individuals. The parameters of the distribution, the vector of means and the covariance matrix, are estimated by means of the maximum likelihood method.

EGNA

The Estimation of Gaussian Network Algorithm (EGNA) by Larrañaga et al. (1999, 2000a) uses a multivariate Gaussian density based on a Bayesian network. A multivariate Gaussian distribution represented by a Bayesian network is called Gaussian network. The EGNA learns the Gaussian factorization of the selected individuals in each generation. Once the network is learnt, it is used to sample new individuals.

2.5 Summary

In this chapter we have presented a brief review of some well known EDAs. According to the domain of decision variables in an optimization problem, the EDAs can be classified as discrete or continuous. An important element that distinguishes EDAs from other EAs is the use of a probabilistic model, which explicitly expresses the interactions among decision variables. In EDAs, the learning of a probabilistic model is based on the data provided by the selected individuals. A general framework for learning a probabilistic model includes the search of a structure (model selection) and the estimation of parameters (model

fitting). For some EDAs, the structure of the probabilistic model is predefined and the only requirement is to estimate (1) an ancestral ordering, and (2) the parameters.

Depending on the domain, several probabilistic models have been incorporated to EDAs. For example, we have presented in this chapter, multinomial distributions, multivariate normal distribution, Bayesian and Markov networks. The learning of the probabilistic model can be based on the use of maximum likelihood or some heuristic that takes into account the fitness values of the best individuals. Thus, according to the probabilistic model used, most of the research on EDAs has been conducted for studying their performance and designing new algorithms.

Chapter 3

Copula Functions

The copula functions are suitable tools in statistics for modeling dependencies, not necessarily linear dependence, in several random variables. The copula theory was introduced by Sklar (1959) to separate the effect of dependence from the effect of marginal distributions in a joint distribution. Although copula functions can model linear and nonlinear dependencies, they have been barely used in computer science applications where nonlinear dependencies are common and need to be represented. In this chapter, we provide an introduction to the copula theory and present several copula functions.

3.1 Introduction

Copula functions have been widely used in economics and finance (Cherubini et al., 2004; Dowd, 2008; Frees and Valdez, 1998; Trivedi and Zimmer, 2007; Venter et al., 2007). More recently copula functions have been used in other fields such as climate (Schölzel and Friederichs, 2008), oceanography (De-Waal and Van-Gelder, 2005), hydrology (Genest and Favre, 2007), geodesy (Bacigál and Komorníková, 2006), reliability (Monjardin, 2007), evolutionary computation (Salinas-Gutiérrez et al., 2009, 2010c, 2011b) and engineering (Grigoriu, 2007). By using copula theory, a joint distribution can be built with a copula function and, possibly, several different marginal distributions. Copula theory has been used also for modeling multivariate distributions in *unsupervised learning* problems such as image segmentation (Brunel et al., 2005; Flitti et al., 2005) and retrieval tasks (Mercier et al., 2007; Sakji-Nsibi and Benazza-Benyahia, 2008; Stitou et al., 2009). In (Jajuga and Papla, 2006), the bivariate Archimedean copula functions Ali-Mikhail-Haq, Clayton, Frank and Gumbel are used for unsupervised classification. These copulas are well defined for two variables but when extended to three or more variables several complications arise, preventing their generalization and applicability. Some of these complications are (1) the copula parameter is the same for all pairs, and (2) it is not possible to model separately the dependence among all pairs of variables. For the Gaussian

copula however, there exist a simple “general formula” for any number of variables. The research works by Salinas-Gutiérrez et al. (2010b,a) introduce the use of Gaussian copula in supervised classification, and compares an independent probabilistic classifier with a copula-based probabilistic classifier.

Definition 3.1. A copula function is a joint distribution function of standard uniform random variables. That is,

$$C(u_1, \dots, u_d) = P[U_1 \leq u_1, \dots, U_d \leq u_d] ,$$

where $U_i \sim U(0, 1)$ for $i = 1, \dots, d$.

As a consequence of Definition 3.1, the copula density for continuous random variables can be calculated as:

$$c(u_1, \dots, u_d) = \frac{\partial^d C(u_1, \dots, u_d)}{\partial u_1 \cdots \partial u_d} . \quad (3.1)$$

The interested reader is referred to (Joe, 1997; Nelsen, 2006; Trivedi and Zimmer, 2007) for a more formal definition of copula function. The following result, known as Sklar’s theorem, gives the relevance and practical utility to copula functions.

Theorem 3.1 (Sklar). *Let F be a d -dimensional distribution function with marginals F_1, F_2, \dots, F_d , then there exists a copula C such that for all x in $\overline{\mathbb{R}}^d$,*

$$F(x_1, x_2, \dots, x_d) = C(F_1(x_1), F_2(x_2), \dots, F_d(x_d)) ,$$

where $\overline{\mathbb{R}}$ denotes the extended real line $[-\infty, \infty]$. If $F_1(x_1), F_2(x_2), \dots, F_d(x_d)$ are all continuous, then C is unique. Otherwise, C is uniquely determined on $\text{Ran}(F_1) \times \text{Ran}(F_2) \times \dots \times \text{Ran}(F_d)$, where Ran stands for the range.

According to Theorem 3.1 and using the chain rule for differentiating composite functions along with the Equation(3.1), any d -dimensional density f can be represented as

$$f(x_1, \dots, x_d) = \prod_{i=1}^d f_i(x_i) \cdot c(F_1(x_1), \dots, F_d(x_d)) , \quad (3.2)$$

where c is the density of the copula C , and $f_i(x_i)$ is the marginal density of variable x_i . The Equation (3.2) shows that the dependence structure is modeled by the copula function. This expression separates any joint density function into the product of copula density and marginal densities. This contrasts with the usual way to model multivariate distributions, which suffers from the restriction that the marginal distributions are usually of the same type. The separation between marginal distributions and a dependence structure explains the modeling flexibility given by copula functions.

3.2 Bivariate copula functions

We start the presentation of the copula theory with an example in the two dimensional case.

Example 3.1. Let $f(x, y) = 4xy [1 + \theta(1 - 2x^2)(1 - 2y^2)]$ be a density function defined over the square with vertices $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$, and where θ is a real number between -1 and 1. The density and distribution function for each random variable along with the joint distribution function can be calculated as follows

$$f(x) = \int_0^1 f(x, y) dy = 2x \quad , \quad (3.3)$$

$$F(x) = \int_0^x 2s ds = x^2 \quad , \quad (3.4)$$

$$f(y) = \int_0^1 f(x, y) dx = 2y \quad , \quad (3.5)$$

$$F(y) = \int_0^y 2t dt = y^2 \quad , \quad (3.6)$$

$$F(x, y) = \int_0^y \int_0^x f(s, t) ds dt = x^2 y^2 [1 + \theta(1 - x^2)(1 - y^2)] \quad . \quad (3.7)$$

As stated below, it is not difficult to see that variables X and Y are not independent

$$f(x) \cdot f(y) = 4xy \neq 4xy [1 + \theta(1 - 2x^2)(1 - 2y^2)] = f(x, y) \quad . \quad (3.8)$$

The result in (3.8) means that there is a statistical dependence between variables X and Y . However, from (3.4), (3.6) and (3.7), we can see the following relationship between the joint distribution function and the marginal distribution functions:

$$F(x, y) = F(x)F(y) [1 + \theta(1 - F(x))(1 - F(y))] \quad . \quad (3.9)$$

Thus, *the joint distribution $F(x, y)$ can be seen as a function of the marginal distributions $F(x)$ and $F(y)$* . This is precisely what Sklar's theorem states for *any* distribution function. Theorem 3.1 provides the theoretical basis for the relationship in (3.9).

Moreover, the joint density $f(x, y)$ can be also seen as a function of the marginal densities and the marginal distributions:

$$\begin{aligned}
 f(x, y) &= \frac{\partial^2 F(x, y)}{\partial x \partial y} \\
 &= f(x)f(y) + \theta f(x)f(y)(1 - 2F(x))(1 - 2F(y)) \\
 &= f(x)f(y) \underbrace{[1 + \theta(1 - 2F(x))(1 - 2F(y))]}_{\text{dependence function}} . \tag{3.10}
 \end{aligned}$$

The right hand of expression (3.10) shows once again that variables X and Y are not independent. Nonetheless, the important result in (3.10) is that the dependence structure of random variables X and Y can be *explicitly* modeled by a function. Equations (3.2) and (3.10) can be compared in order to know the copula density associated to the joint density $f(x, y)$.

To conclude with this example, it can be noted that the parameter θ plays an important role for describing the association between variables X and Y . This parameter only appears in the joint functions and does not appear in the marginal functions. When $\theta = 0$, the *dependence function* in (3.10) is equal to one and the variables X and Y are independent. ◀

An explicit expression for the dependence between variables could be useful for computational and statistical purposes such as inference, modeling, simulation, and measuring the association.

3.2.1 Fréchet-Hoeffding bounds and the product copula

Three important cases of bivariate copula functions are the Fréchet-Hoeffding lower bound $W(u, v)$, the Fréchet-Hoeffding upper bound $M(u, v)$, and the product copula $\Pi(u, v)$. Fréchet-Hoeffding bounds describe complete negative and positive dependence, whereas the product copula is related to independent variables.

The mathematical definition for these copula functions is the following:

$$W(u, v) = \max(u + v - 1, 0) , \tag{3.11}$$

$$M(u, v) = \min(u, v) , \tag{3.12}$$

$$\Pi(u, v) = uv . \tag{3.13}$$

In Figure 3.1, the Fréchet-Hoeffding copulas and the product copula are shown. It can be proven, see Nelsen (2006), that for any copula C ,

$$W(u, v) \leq C(u, v) \leq M(u, v) , \tag{3.14}$$

The result in (3.14) explains why the Fréchet-Hoeffding copulas are called *bounds*. Moreover, it means that the graph of any copula function is bounded

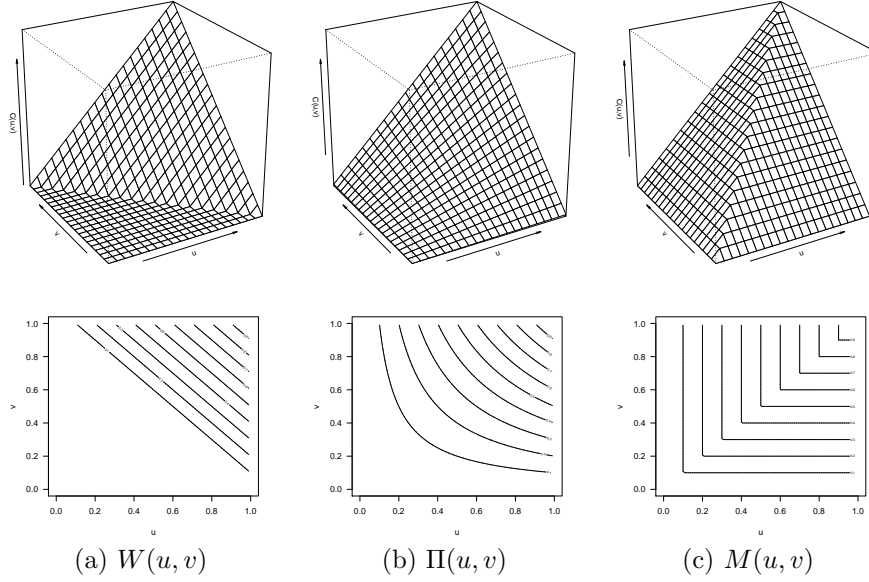


Figure 3.1: Graphs and contour diagrams of (a) the Fréchet-Hoeffding lower bound, (b) the product copula, and (c) the Fréchet-Hoeffding upper bound.

below and above by the graph of the Fréchet-Hoeffding lower bound and the graph of the Fréchet-Hoeffding upper bound.

The Fréchet-Hoeffding lower bound and the Fréchet-Hoeffding upper bound are also known as the *minimum copula* and the *maximum copula* (Cherubini et al., 2004). A family of copulas that includes M , Π , and M is called *comprehensive* (Nelsen, 2006).

3.2.2 Some bivariate copulas

Farlie-Gumbel-Morgenstern copula

The Farlie-Gumbel-Morgenstern (FGM) copula can be seen as a slight modification of the product copula (3.13). This copula function is defined as follows:

$$C(u, v) = uv(1 + \theta(1 - u)(1 - v)) \quad (3.15)$$

The copula parameter θ in (3.15) takes values in the interval $[-1, 1]$. It can be noted that, if θ equals zero, the independence case is considered by the FGM copula function.

According to expression (3.1), the density function for the FGM copula can be calculated by differentiating twice the distribution function (3.15)

$$c(u, v) = 1 + \theta(1 - 2u)(1 - 2v) \quad (3.16)$$

Figure 3.2 (a) shows the graph of a FGM copula distribution and a FGM copula density for a negative dependence between variables.

The FGM copula was introduced in Example 3.1. Equations (3.9) and (3.10) can be compared to the FGM copula distribution (3.15) and the FGM copula density (3.16), respectively.

Frank copula

The Frank copula is defined by the following expression

$$C(u, v) = -\frac{1}{\theta} \ln \left(1 + \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1} \right) . \quad (3.17)$$

The dependence parameter θ in (3.17) may take any real value except the zero. The Frank copula is considered a comprehensive family because the limiting cases $\theta \rightarrow \pm\infty$, and $\theta \rightarrow 0$ include the Fréchet-Hoeffding bounds and the product copula.

The density function for the Frank copula is given by

$$c(u, v) = \frac{-\theta(e^{-\theta} - 1)e^{-\theta(u+v)}}{((e^{-\theta u} - 1)(e^{-\theta v} - 1) + (e^{-\theta} - 1))^2} . \quad (3.18)$$

The Frank copula is mostly appropriate for data that exhibit weak dependence between extreme values and strong dependence between centered values (Trivedi and Zimmer, 2007). Figure 3.2 (b) shows the graph of a Frank copula distribution and a Frank copula density for a positive dependence between variables. The Frank copula belongs to an important class of copulas known as Archimedean copulas.

Gaussian copula

This copula function is based on the bivariate standard normal distribution and hence, it does not have a closed form as the previous copula functions. The Gaussian copula distribution and the Gaussian copula density are given by the following expressions:

$$C(u, v) = \int_{-\infty}^{\Phi^{-1}(u)} \int_{-\infty}^{\Phi^{-1}(v)} \frac{e^{-\frac{1}{2}t'\Sigma^{-1}t}}{2\pi|\Sigma|^{1/2}} dt_1 dt_2 , \quad (3.19)$$

$$c(u, v) = (1 - \theta^2)^{1/2} \exp \left(-\frac{(x^2 + y^2 - 2\theta xy)}{2(1 - \theta^2)} + \frac{(x^2 + y^2)}{2} \right) , \quad (3.20)$$

where Σ is a correlation matrix with $\Sigma_{12} = \theta \in (-1, 1)$, Φ is the cumulative distribution function of the marginal standard normal distribution, $x = \Phi^{-1}(u)$, and $y = \Phi^{-1}(v)$. As the copula parameter θ approaches -1 and 1 , the Gaussian copula attains the Fréchet-Hoeffding lower and upper bound, respectively.

Similar to the Frank copula, the Gaussian copula permits negative and positive dependence between the marginals. Moreover, the Gaussian copula is also a comprehensive family in the sense that both Fréchet-Hoeffding bounds and the product copula are included in the range of permissible dependence. However, unlike the Frank copula, this copula function is appropriate for data that exhibit weak dependence between centered values and strong dependence between extreme values.

The Gaussian copula is a good example of a copula function defined in terms of a very well known distribution. Not surprising, the estimation of the copula parameter θ and the sampling procedure for the Gaussian copula is also based on the estimation and sampling methods of the bivariate standard normal distribution.

Figure 3.2 (c) shows the graph of a Gaussian copula distribution and a Gaussian copula density for a negative dependence between variables. The Gaussian copula belongs to the class of elliptical copulas.

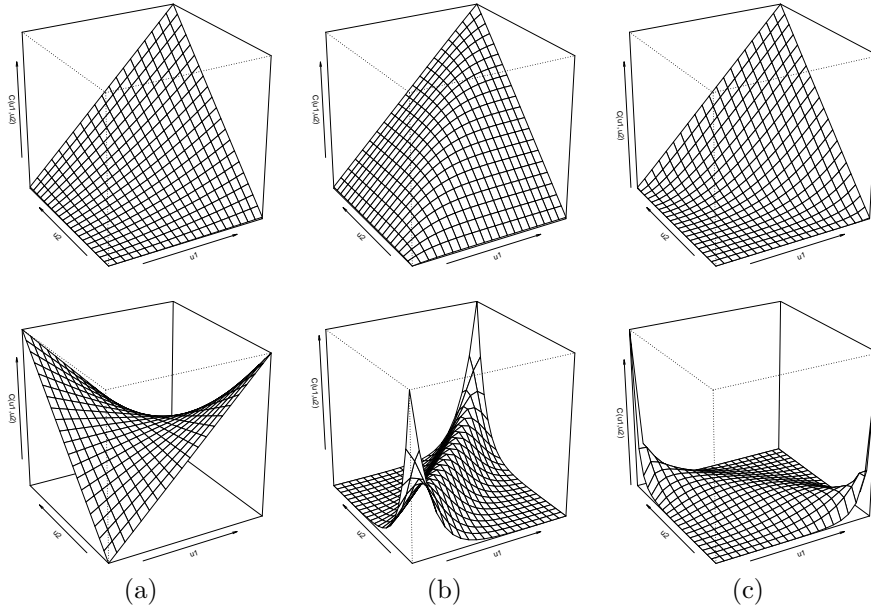


Figure 3.2: Distribution (top) and density (bottom) functions for (a) the FGM copula with $\theta = -0.8$, (b) the Frank copula with $\theta = 10$, and (c) the Gaussian copula with $\theta = -0.8$.

3.2.3 Estimation of the parameter

For parametric bivariate copula functions, Kendall's correlation rank can be expressed in terms of the copula parameter (see Nelsen, 2006).

Theorem 3.2. *Let X and Y be continuous random variables whose copula is C . Then the population version of Kendall's tau for X and Y is given by*

$$\tau(X, Y) = 4 \int_0^1 \int_0^1 C(u, v; \theta) dC(u, v; \theta) - 1, \quad (3.21)$$

where $u = F_X(x)$ and $v = F_Y(y)$.

Theorem 3.2 states a relationship between the copula parameter θ and the measure of association Kendall's tau. For this reason, given a nonparametric estimation of Kendall's tau from data $\{(x_i, y_i)\}_{i=1}^n$, the relation (3.21) can be used for estimating the dependence parameter θ . Moreover, given that Kendall's tau is invariant under continuous and increasing transformations, the nonparametric estimation of Kendall's tau can be also calculated from the transformed data $\{(u_i, v_i)\}_{i=1}^n$, where $u_i = F_X(x_i)$ and $v_i = F_Y(y_i)$. However, for this thesis, the nonparametric estimation of the copula parameter via Kendall's tau, is only the initial step for learning such parameter.

The dependence parameter θ of a bivariate copula function can be estimated using the maximum likelihood method along with the nonparametric estimation of Kendall's tau. To do so, the one-dimensional log-likelihood function

$$\ell(\theta; \{(u_i, v_i)\}_{i=1}^n) = \sum_{i=1}^n \ln c(u_i, v_i; \theta), \quad (3.22)$$

is maximized using as starting point the nonparametric estimation of Kendall's tau. In this thesis the copula parameters are estimated via the maximum likelihood method. It has been shown in (Weiß, 2011) that the maximum likelihood estimator has better properties than other estimators.

3.2.4 Sampling from a copula function

In general, there are several methods for making pseudo-random draws from a bivariate distribution, for example, the conditional sampling method and other Monte Carlo methods. In the case of a bivariate copula function these methods can be also used for generating samples. However, in order to simulate data in EDAs, a convenient method used in this thesis is the conditional sampling.

For sampling from bivariate copula functions, the Algorithm 2 gives the steps of simulating variates.

We explain the application of Algorithm 2 in a practical example.

Example 3.2. Consider the simulation of the Frank copula for different values of the dependence parameter θ . For applying Algorithm 2, it is necessary to calculate the partial derivative of the Frank copula respect to the variable u :

Algorithm 2 Pseudocode for generating samples from a bivariate copula function

- 1: Draw two independent random variables (u, t) from an uniform distribution $U(0, 1)$.
- 2: Solve $t = C_v(v|u; \theta)$ for v , where $C_v(v|u; \theta) = \frac{\partial C}{\partial u}$ is the conditional distribution of V given U .
The pair (u, v) is a simulation from the bivariate copula function $C(u, v; \theta)$.

$$\frac{\partial C}{\partial u} = \frac{e^{-\theta u}(e^{-\theta v} - 1)}{(e^{-\theta u} - 1)(e^{-\theta v} - 1) + (e^{-\theta} - 1)} . \quad (3.23)$$

Once the conditional distribution of V given U is calculated (3.23), the next step consists in solving the following equation for v :

$$t = \frac{e^{-\theta u}(e^{-\theta v} - 1)}{(e^{-\theta u} - 1)(e^{-\theta v} - 1) + (e^{-\theta} - 1)} , \quad (3.24)$$

where t is a number in the interval $(0, 1)$.

The solution for v is given by the expression

$$v = -\frac{1}{\theta} \ln \left(\frac{(1-t)e^{-\theta u} + te^{-\theta}}{(1-t)e^{-\theta u} + t} \right) . \quad (3.25)$$

Therefore, a simulation of the Frank copula follows the next steps

1. Draw an observation of the random variable U by sampling from the uniform distribution $U(0, 1)$. For getting an observation of the random variable V , it is necessary to have an independent auxiliary sample t from the uniform distribution $U(0, 1)$.
2. The observation of the random variable V is made by means of the expression (3.25) and the values of (u, t) .

The example is concluded by showing in Figure 3.3 two data sets simulated from the Frank copula.



A selected number of bivariate copula are presented in the Appendix A.

3.3 Multivariate copulas

Two recognized groups of copula functions in the literature are the elliptical class and the Archimedean class. Some bivariate copula functions of these classes have already been introduced in the previous section.

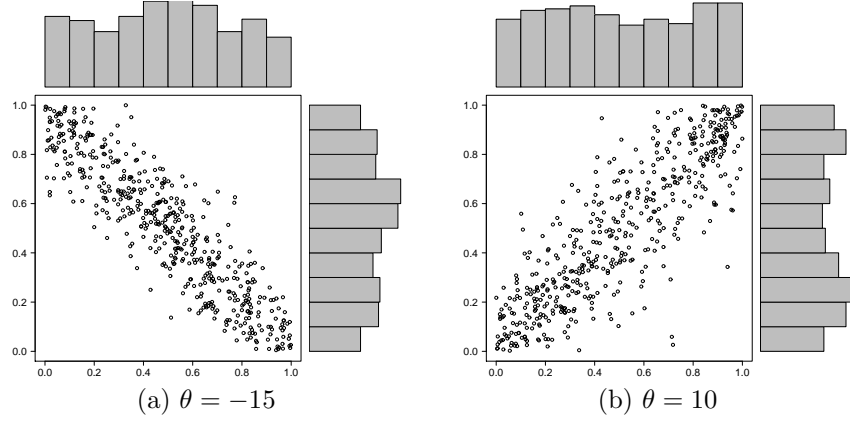


Figure 3.3: Two samples of 500 points each from the Frank copula function for (a) a negative dependence, and (b) a positive dependence. All marginal distributions are uniform.

3.3.1 The Gaussian copula

An important parametric family is the multivariate Gaussian copula. This copula function along with the multivariate Student's copula are members of the elliptical copulas.

Definition 3.2. The copula associated to the multivariate standard normal distribution is called Gaussian copula.

According to Definition 3.2 and Theorem 3.1, if the d -dimensional distribution of a random vector (Z_1, \dots, Z_d) is a joint standard normal distribution, then the associated Gaussian copula has the following expression:

$$C(\Phi(z_1), \dots, \Phi(z_d); \Sigma) = \int_{-\infty}^{z_1} \cdots \int_{-\infty}^{z_d} \frac{e^{-\frac{1}{2}t'\Sigma^{-1}t}}{(2\pi)^{(n/2)}|\Sigma|^{1/2}} dt_d \cdots dt_1, \quad (3.26)$$

or equivalently,

$$C(u_1, \dots, u_d; \Sigma) = \int_{-\infty}^{\Phi^{-1}(u_1)} \cdots \int_{-\infty}^{\Phi^{-1}(u_d)} \frac{e^{-\frac{1}{2}t'\Sigma^{-1}t}}{(2\pi)^{(n/2)}|\Sigma|^{1/2}} dt_d \cdots dt_1, \quad (3.27)$$

where Φ is the cumulative distribution function of the marginal standard normal distribution and Σ is a symmetric matrix with main diagonal of ones. The elements outside the main diagonal of matrix Σ are the pairwise correlations ρ_{ij} between variables Z_i and Z_j , for $i, j = 1, \dots, d$ and $i \neq j$. It can be noticed that a d -dimensional standard normal distribution has mean vector zero and a correlation matrix Σ with $d(d-1)/2$ parameters.

The dependence parameters ρ_{ij} of a d -dimensional Gaussian copula can be estimated using the maximum likelihood method. To do so, the steps of Algorithm 3 can be followed. Algorithm 3 is based on reference (Cherubini et al., 2004).

Algorithm 3 Pseudocode for estimating Gaussian copula parameters

- 1: Transform values of each variable u_j by calculating $z_j = \Phi^{-1}(u_j)$, for $j = 1, \dots, d$, where Φ is the cumulative standard normal distribution function.
- 2: Build the sample data matrix $\mathbf{z} = \{(z_{1i}, z_{2i}, \dots, z_{di})\}_{i=1}^n$.
- 3: Estimate the correlation matrix $\widehat{\Sigma}$ using pseudo observations $\mathbf{z}_i = (z_{1i}, z_{2i}, \dots, z_{di})$ and the formula

$$\widehat{\Sigma} = \frac{1}{n} \sum_{i=1}^n \mathbf{z}'_i \mathbf{z}_i .$$

Due to Equation (3.1), the d -dimensional Gaussian copula density can be calculated as:

$$\begin{aligned} c(\Phi(z_1), \dots, \Phi(z_d); \Sigma) &= \frac{1}{(2\pi)^{(d/2)} |\Sigma|^{1/2}} e^{-\frac{1}{2} \mathbf{z}' \Sigma^{-1} \mathbf{z}} \\ &= \frac{1}{\prod_{i=1}^d \frac{1}{(2\pi)^{1/2}} e^{-\frac{1}{2} z_i^2}} \\ &= \frac{1}{|\Sigma|^{1/2}} e^{-\frac{1}{2} \mathbf{z}' (\Sigma^{-1} - I) \mathbf{z}} . \end{aligned} \quad (3.28)$$

Given that a Gaussian copula is also a distribution function, it is possible to simulate data from it. The main steps are the following: once a correlation matrix Σ is specified, a data set can be generated from a joint standard normal distribution. The next step consists of transforming this data set using the cumulative distribution function Φ . Algorithm 4 and Figure 3.4 illustrate the sampling procedure for different correlations.

Algorithm 4 Pseudocode for generating data from a Gaussian copula

- 1: Simulate observations (z_1, \dots, z_d) from a joint standard normal distribution with correlation matrix Σ .
 - 2: Calculate $u_i = \Phi(z_i)$ where Φ is the cumulative standard normal distribution function, for $i = 1, \dots, d$.
-

Figure 3.4 (a)-(top) shows a sample drawn from a bivariate standard normal distribution with correlation $\rho = -0.70$ (step 1, Algorithm 4). The histogram on the vertical axis and the histogram on the horizontal axis illustrate that both marginals are univariate standard normal distributions. This data set is used to obtain a sample from a Gaussian copula, as shown in Figure 3.4 (a)-(bottom) (step 2, Algorithm 4). Both histograms illustrate that marginals are uniform, according to Definition 3.1. In order to appreciate how the correlation parame-

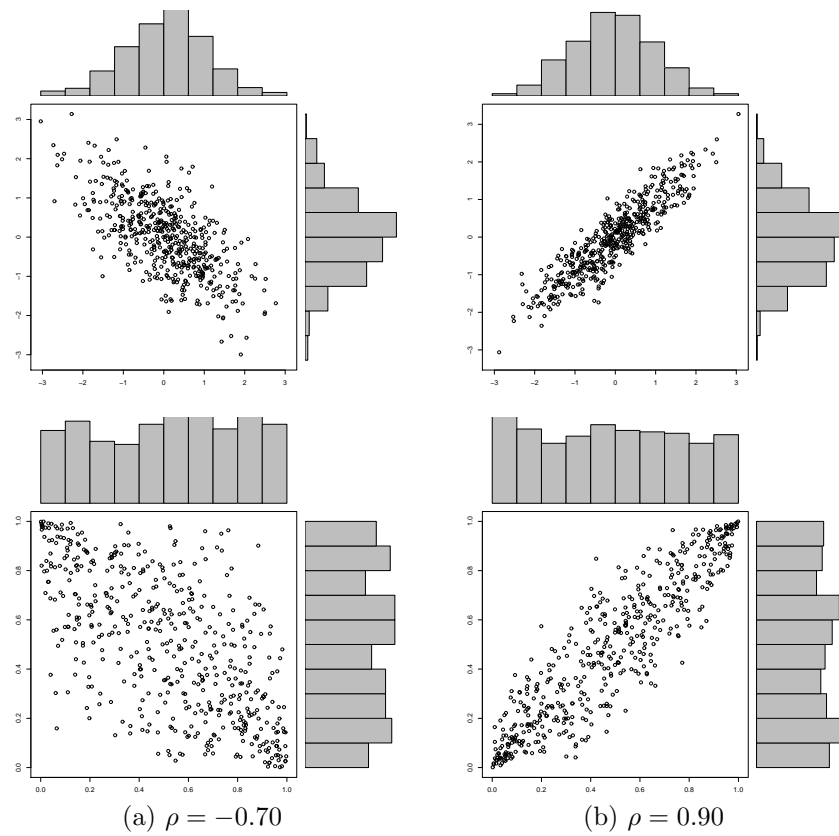


Figure 3.4: A sample of 500 points from a standard normal distribution (top) and the corresponding sample for a Gaussian copula (bottom) with (a) a negative dependence, and (b) a positive dependence. Histograms show the marginal distribution for each variable.

ter modifies the dependence structure, Figure 3.4 (b) shows the corresponding information with $\rho = 0.90$.

An important result for parametric bivariate copulas is explained through Equation (3.21), which relates the dependence parameter θ of a copula and Kendall's τ . For a bivariate Gaussian copula, Equation (3.21) can be written as

$$\tau = \frac{2}{\pi} \arcsin(\rho) . \quad (3.29)$$

As a consequence, the data sets in Figure 3.4 (a)-(top,bottom) have the same concordance value, measured in Kendall's τ . A similar statement can be said for Figure 3.4 (b)-(top,bottom).

Given that is well established how to estimate correlation matrices, evaluate densities, and calculate integrals for the multidimensional normal distribution, the Gaussian copula function is relatively easy to implement.

3.3.2 Archimedean copulas

Another important class of copulas are the Archimedean copulas. These copulas are popular because they are easily constructed and are able of capturing wide ranges of dependence.

The construction of an Archimedean copula is based on the definition of a *generator function* φ . A function φ is called *generator* if it satisfies the following conditions

- $\varphi(t) : [0, 1] \rightarrow [0, \infty]$
- φ is continuous
- φ is strictly decreasing, i.e. $\varphi'(t) < 0$ for all $0 < t < 1$
- φ is convex, i.e. $\varphi''(t) > 0$ for all $0 < t < 1$
- $\varphi(1) = 0$

Any function φ that satisfied these properties is capable of generating an Archimedean copula. The above conditions are necessary for ensuring the existence of the *pseudo-inverse* $\varphi^{[-1]}$. The pseudo-inverse of φ is the function given by

$$\varphi^{[-1]}(t) = \begin{cases} \varphi^{-1}(t) & 0 \leq t \leq \varphi(0) \\ 0 & \varphi(0) \leq t \leq \infty \end{cases} \quad (3.30)$$

It can be noticed that the pseudo-inverse $\varphi^{[-1]}$ is continuous and nonincreasing on $[0, \infty]$, and strictly decreasing on $[0, \varphi(0)]$. Moreover, the composition of the pseudo-inverse with the generator gives the identity, $\varphi^{[-1]}(\varphi(t)) = t$.

When the additional condition $\varphi(0) = \infty$ is also satisfied, the pseudo-inverse of φ coincides with the usual inverse $\varphi^{[-1]} = \varphi^{-1}$, and the function φ is said to be a *strict generator*.

The following theorem gives the necessary and sufficient conditions for generating multivariate Archimedean copulas.

Theorem 3.3. *Let φ be a continuous, strictly decreasing function from $[0, 1]$ to $[0, \infty]$ such that $\varphi(0) = \infty$ and $\varphi(1) = 0$, and let φ^{-1} denote the inverse of φ . If C is the function from $[0, 1]^d$ to $[0, 1]$ given by*

$$C(u_1, u_2, \dots, u_d) = \varphi^{-1}(\varphi(u_1) + \varphi(u_2) + \dots + \varphi(u_d)) \quad , \quad (3.31)$$

then C is a multivariate copula for all $d \geq 2$ if and only if φ^{-1} is completely monotonic on $[0, \infty)$.

Theorem 3.3 requires a strict generator φ and a completely monotonic inverse φ^{-1} for defining a multivariate Archimedean copula. A function $g(t)$ is called *completely monotonic* on an interval I if it is continuous there and has derivatives of all orders that alternate in sign, i.e.,

$$(-1)^k \frac{d^k}{dt^k} g(t) \geq 0 \quad , \quad (3.32)$$

for all t in the interior of I and $k \in \{0, 1, 2, \dots\}$.

We present two multivariate Archimedean copulas.

Frank copula

The generator function and its inverse are given by

$$\varphi(t) = -\ln \left(\frac{e^{-\theta t} - 1}{e^{-\theta} - 1} \right) \quad , \quad (3.33)$$

$$\varphi^{-1}(s) = -\frac{1}{\theta} \ln \left(1 + e^{-s} (e^{-\theta} - 1) \right) \quad , \quad (3.34)$$

where $\theta > 0$ in order to make $\varphi^{-1}(s)$ completely monotonic. The multivariate Frank copula is given by

$$C(u_1, u_2, \dots, u_d) = -\frac{1}{\theta} \ln \left(1 + \frac{\prod_{i=1}^d (e^{-\theta u_i} - 1)}{(e^{-\theta} - 1)^{d-1}} \right) \quad , \quad (3.35)$$

with $\theta > 0$ when $d \geq 3$. For $d = 2$, θ can also take values less than zero.

Gumbel copula

The generator for the Gumbel copula is given by

$$\varphi(t) = (-\ln(t))^\theta \quad , \quad (3.36)$$

and the inverse

$$\varphi^{-1}(s) = e^{-s^{1/\theta}} \quad . \quad (3.37)$$

The inverse is completely monotonic if $\theta > 1$. The multivariate Gumbel copula is therefore

$$C(u_1, u_2, \dots, u_d) = \exp \left(- \left[\sum_{i=1}^d (-\ln u_i)^\theta \right]^{1/\theta} \right), \quad (3.38)$$

with $\theta > 1$ when $d \geq 3$. For $d = 2$, θ can also include the value one.

The two dimensional case

Less constraints of the generator function φ are required for defining bivariate Archimedean copulas.

Theorem 3.4. *Let φ be a continuous, strictly decreasing function from $[0, 1]$ to $[0, \infty]$ such that $\varphi(1) = 0$, and let $\varphi^{[-1]}$ be the pseudo-inverse of φ defined by (3.30). Then the function C from $[0, 1]^2$ to $[0, 1]$ given by*

$$C(u, v) = \varphi^{[-1]}(\varphi(u) + \varphi(v)), \quad (3.39)$$

is a bivariate copula if and only if φ is convex.

The proof of Theorem 3.4 can be seen in (Nelsen, 2006). It can be noticed in Theorem 3.4 that it is not necessary to use a strict generator function φ , nor a completely monotonic inverse φ^{-1} .

Some of the copula functions used in this thesis are bivariate Archimedean copulas. A description of their generator function φ is shown in Table 3.1 along with their relationship with the Fréchet-Hoeffding and the product copulas.

More details about the bivariate copulas in Table 3.1 are presented in the Appendix A.

3.4 Summary

In this chapter we have shown that the copula theory is related to the study of dependencies among random variables and that the copula functions are probability distribution functions. An important characteristic of copula functions is their theoretical support for modeling *any* kind of dependence among random variables. However, for application purposes, the user must be able to select the adequate copula function according to the information provided by the data.

Methods for estimating the copula parameters have been presented in this chapter. Furthermore, we have also presented methods for sampling from copula functions. In this dissertation, the proposed EDAs estimate the copula parameters by using the maximum likelihood method. The conditional sampling method is used to simulate new individuals.

Table 3.1: Bivariate Archimedean copulas.

Generator $\varphi(t)$	Range of θ	Strict	Limiting and special cases
Ali-Mikhail-Haq			
$\ln \frac{1 - \theta(1-t)}{t}$	$[-1, 1)$	yes	$C_0 = \Pi$
Clayton			
$\frac{1}{\theta} (t^{-\theta} - 1)$	$[-1, \infty) \setminus \{0\}$	$\theta \geq 0$	$C_{-1} = W$ $C_0 = \Pi$ $C_\infty = M$
Frank			
$-\ln \left(\frac{e^{-\theta t} - 1}{e^{-\theta} - 1} \right)$	$(-\infty, \infty) \setminus \{0\}$	yes	$C_{-\infty} = W$ $C_0 = \Pi$ $C_\infty = M$
Gumbel			
$(-\ln(t))^\theta$	$[1, \infty)$	yes	$C_1 = \Pi$ $C_\infty = M$

Chapter 4

EDAs and Copula Functions

This chapter presents the incorporation and use of copula functions for modeling dependencies among continuous variables in Estimation of Distribution Algorithms (EDAs). We show how copula functions can be used along with graphical models for defining new EDAs. Moreover, this chapter also show how the use of mutual information in the learning of graphical models implies a natural way of employing copula functions.

4.1 Introduction

The research on EDAs has been conducted in proposing and enhancing probabilistic models. Most of the probabilistic models used in EDAs for discrete domains are based on graphical models such as Bayesian and Markov Networks (De Bonet et al., 1997; Baluja and Davies, 1997; Soto et al., 1999; Pelikan and Mühlenbein, 1999; Pelikan et al., 1999; Santana-Hermida, 2004; Shakya, 2006). The discrete EDAs have been widely studied and they have been also used in several applications. However, for continuous domains, the Gaussian distribution is the most used assumption over the probabilistic model (Larrañaga et al., 1999, 2000a, 2001; Bosman, 2003). For this reason, one of the current challenges for designing new continuous EDAs is finding multivariate models to adequately represent dependencies among the decision variables.

On the other hand, during the last decade, the copula functions have been widely used in many applications where nonlinear dependencies arise among variables. By using copula functions it is possible to separate the effect of dependence from the effect of marginal distributions in a joint distribution. However, they have been barely used in computer science applications where nonlinear dependencies are common and need to be represented.

One motivation for this research has been the possibility of proposing new EDAs for continuous domains. The copula based EDAs are able, at the same

time, (1) to use flexible marginal distributions, and (2) to model dependencies among variables. As we have presented in the previous chapter, the copula theory gives the means for getting such EDAs.

4.2 Copula functions and graphical models

Despite copula functions can model dependencies among all pairwise variables sometimes it is not clear what multivariate copula function must be chosen. However, by means of graphical models (Whittaker, 1990; Lauritzen, 1996) it is possible to model the most important dependencies or associations between variables. This is the case for a copula function, because a multivariate copula function is also a probabilistic model. We propose in this thesis to use directed acyclic graphs in EDAs as graphical models for multivariate copula functions. Two of these models are illustrated in Figure 4.1 and are based on pairwise conditional distributions.

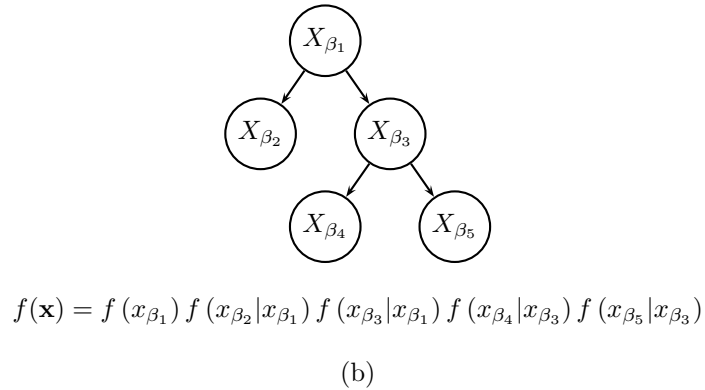
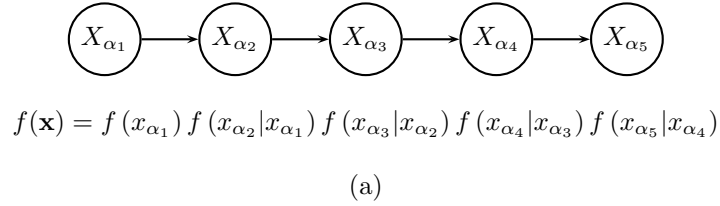


Figure 4.1: Two joint distributions over five variables represented by (a) a chain graphical model and (b) a tree graphical model.

4.2.1 The chain graphical model

A chain graphical model for a d -dimensional continuous random vector \mathbf{X} represents a probabilistic model with the following density:

$$f_{\text{chain}}(\mathbf{x}) = f(x_{\alpha_1}) \prod_{i=2}^d f(x_{\alpha_i} | x_{\alpha_{(i-1)}}) , \quad (4.1)$$

where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d)$ is a permutation of the integers between 1 and d . Figure 4.1 (a) shows an example of a chain graphical model.

In practice the permutation $\boldsymbol{\alpha}$ is unknown and the chain graphical model must be learnt from data. A way of choosing the permutation $\boldsymbol{\alpha}$ is based on the Kullback-Leibler divergence. This divergence is an information measure between two distributions. It is always non-negative for any two distributions, and is zero if and only if the distributions are identical. Hence, the Kullback-Leibler divergence can be interpreted as a measure of the dissimilarity of two distributions. Then, the goal is to choose a permutation $\boldsymbol{\alpha}$ that minimizes the Kullback-Leibler divergence between the true distribution $f(\mathbf{x})$ of the data set and the distribution associated to a chain model, $f_{\text{chain}}(\mathbf{x})$. In this case the Kullback-Leibler divergence, D_{KL} , of a continuous random vector \mathbf{X} with joint densities f and f_{chain} is given by:

$$\begin{aligned} D_{KL}(f || f_{\text{chain}}) &= E_{f(\mathbf{x})} \left[\log \frac{f(\mathbf{x})}{f_{\text{chain}}(\mathbf{x})} \right] \\ &= -H(\mathbf{X}) + H(X_{\alpha_1}) + \sum_{i=2}^d H(X_{\alpha_i} | X_{\alpha_{(i-1)}}) . \quad (4.2) \end{aligned}$$

The first term in the divergence (4.2) is the entropy of the joint distribution $f(\mathbf{x})$ and does not depend on the permutation $\boldsymbol{\alpha}$. The second and third terms are taken into account for minimizing the Kullback-Leibler divergence. However, it can be noticed that there is not a direct solution without considering all possible permutations. For this reason it is necessary to employ a greedy algorithm in order to find a sub-optimal permutation $\boldsymbol{\alpha}$.

In the EDA literature, the algorithm that uses a chain graphical model is the Mutual Information Maximizing Input Clustering (MIMIC). This algorithm proposes a way to find a permutation in the following way: 1) set as root the variable with the lowest marginal entropy, and 2) choose the variable whose conditional entropy with respect to the previous variable is the lowest. The chain is built by repeating the previous step with the rest of the variables.

The MIMIC algorithm for continuous variables uses bivariate Gaussian distributions. This assumption gives a direct way of calculating marginal and conditional entropies, however it can not be realistic for some optimization problems.

In this work we employ an equivalent expression for minimizing the Kullback-

Leibler divergence (4.2). It is well known in information theory (Cover and Thomas, 1991) that the conditional entropy between variables S and T is related to its mutual information $I(S, T)$ as follows:

$$H(T|S) = H(T) - I(S, T) \quad , \quad (4.3)$$

where

$$I(S, T) = E_{f(s,t)} \left[\log \frac{f(s,t)}{f(s) \cdot f(t)} \right] \quad .$$

Substituting conditional entropies in Equation (4.2) by the relation given by Equation (4.3), the Kullback-Leibler divergence can be written as:

$$D_{KL}(f||f_{\text{chain}}) = -H(\mathbf{X}) + \sum_{k=1}^d H(X_k) - \sum_{i=2}^d I(X_{\alpha_i}, X_{\alpha_{(i-1)}}) \quad . \quad (4.4)$$

According to Equation (4.4), minimizing the Kullback-Leibler divergence is equivalent to maximizing the total sum:

$$J_{\text{chain}}(\mathbf{X}) = \sum_{i=2}^d I(X_{\alpha_i}, X_{\alpha_{(i-1)}}) \quad . \quad (4.5)$$

A sub-optimal permutation α can be found by taking into account just the total sum of pairwise mutual information. As we shall show later, the entropy of the copula function gives a natural way for computing the Equation (4.5).

4.2.2 The tree graphical model

The joint density for a tree graphical model is given by:

$$f_{\text{tree}}(\mathbf{x}) = f(x_{\beta_1}) \prod_{i=2}^d f(x_{\beta_i} | x_{\beta_{p(i)}}) \quad , \quad (4.6)$$

where $\beta = (\beta_1, \dots, \beta_d)$ is a permutation of integers $1, \dots, d$, and $p(i)$ maps numbers $1 < i \leq d$ to integers $1 \leq p(i) < i$. Each variable in Equation (4.6) is conditioned upon at most one parent. It can be noticed that the chain graphical model is a special case of the tree graphical model when $p(i) = i - 1$. Figure 4.1 (b) shows an example of a tree graphical model.

In a similar way to the learning of a chain graphical model, a tree model can be learned by minimizing the Kullback-Leibler divergence between the true density function $f(\mathbf{x})$ and the proposed density function, $f_{\text{tree}}(\mathbf{x})$:

$$D_{KL}(f||f_{\text{tree}}) = -H(\mathbf{X}) + \sum_{k=1}^d H(X_k) - \sum_{i=2}^d I(X_{\beta_i}, X_{\beta_{p(i)}}) \quad . \quad (4.7)$$

The first two terms in Equation (4.7) are entropies and do not depend on the

tree graphical structure. The third term is the sum of all mutual information associated to the $d-1$ tree branches. Therefore, minimizing the Kullback-Leibler divergence is equivalent to maximizing the total sum:

$$J_{\text{tree}}(\mathbf{X}) = \sum_{i=2}^d I(X_{\beta_i}, X_{\beta_{p(i)}}) . \quad (4.8)$$

The optimization problem (4.8) was presented in (Chow and Liu, 1968) as the total branch weight and it has been used for the learning of dependence trees. The optimal tree is the one that produces the highest total sum of mutual information. Since the amount of mutual information for every pair of nodes is independent of the rest, Kruskal's algorithm can be used for maximizing the summation in Equation (4.8).

4.2.3 Copula entropy and mutual information

By using copula functions, see Equation (3.1), the joint density represented by a chain model can be written as

$$\begin{aligned} f_{\text{chain}}(\mathbf{x}) &= f(x_{\alpha_1}) \prod_{i=2}^d f(x_{\alpha_i} | x_{\alpha_{(i-1)}}) \\ &= f(x_{\alpha_1}) \prod_{i=2}^d \frac{f(x_{\alpha_i}) \cdot f(x_{\alpha_{(i-1)}}) \cdot c(u_{\alpha_i}, u_{\alpha_{(i-1)}})}{f(x_{\alpha_{(i-1)}})} \\ &= \prod_{k=1}^d f(x_k) \prod_{i=2}^d c(u_{\alpha_i}, u_{\alpha_{(i-1)}}) . \end{aligned} \quad (4.9)$$

The above result is very illustrative. From Sklar's theorem we know that there is an unique copula function which models the dependence structure. The equation (4.9) states that the copula density associated to a chain graphical model, $c_{\text{chain}}(\mathbf{u})$, can be decompose into the product of bivariate copula functions:

$$c_{\text{chain}}(\mathbf{u}) = \prod_{i=2}^d c(u_{\alpha_i}, u_{\alpha_{(i-1)}}) . \quad (4.10)$$

This is an interesting and practical consequence, because *any* multivariate copula function associated to a chain model can be built by employing the adequate bivariate copula functions.

Moreover, the copula function associated to the chain model can be also

graphically represented by a chain. To see that,

$$\begin{aligned} c_{\text{chain}}(\mathbf{u}) &= \prod_{i=2}^d c(u_{\alpha_i}, u_{\alpha_{(i-1)}}) \\ &= \prod_{i=2}^d c(u_{\alpha_i} | u_{\alpha_{(i-1)}}) . \end{aligned} \quad (4.11)$$

The previous calculations used the fact that the marginal distributions of a copula function are uniform distributions. It can be noted that Equation (4.11) is similar to Equation (4.1).

For the case of a tree model it is not difficult to see the following:

$$f_{\text{tree}}(\mathbf{x}) = c_{\text{tree}}(\mathbf{u}) \prod_{k=1}^d f(x_k) , \quad (4.12)$$

where

$$c_{\text{tree}}(\mathbf{u}) = \prod_{i=2}^d c(u_{\beta_i}, u_{\beta_{p(i)}}) . \quad (4.13)$$

The result in Equation 4.13 states that the copula function associated to a joint distribution with a tree graphical model is also represented by a tree. Figure 4.2 illustrates the results given by Equations (4.9) to (4.13) for a joint distribution with five variables.

Next, we discuss how the learning of a chain model or a tree model for the random vector \mathbf{X} is related to the entropy of a copula function. We have seen that picking the sub optimal permutation for a chain model and the optimal permutation for a tree model depends on the total sum of pairwise mutual information, Equations (4.5) and (4.8).

In this thesis we propose the use of the following relationship, presented in (Davy and Doucet, 2003), between the mutual information of variables (S, T) and the entropy of their associated copula function:

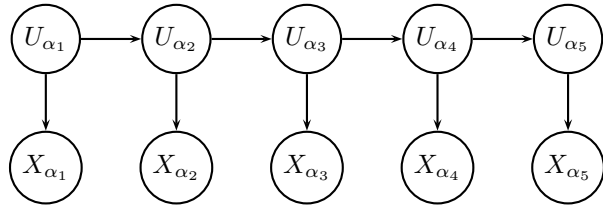
$$I(S, T) = -H(U, V) , \quad (4.14)$$

where copula variables (U, V) are related to variables (S, T) by their marginal distribution function, i.e., $u = F_S(s)$ and $v = F_T(t)$.

Therefore, the Equation (4.5) can be written as:

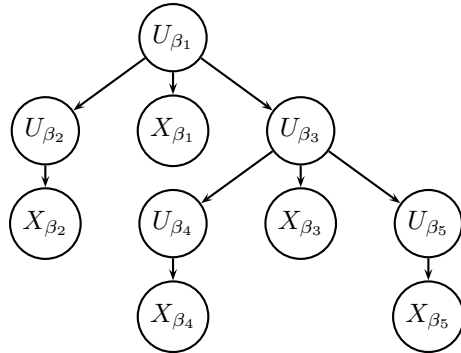
$$J_{\text{chain}}(\mathbf{X}) = - \sum_{i=2}^d H(U_{\alpha_i}, U_{\alpha_{(i-1)}}) . \quad (4.15)$$

A consequence of Equation (4.15) is that a sub optimal permutation α for a chain model can be learned by using the sum of bivariate copula entropies instead of the sum of pairwise mutual information. This can be taken into account



$$f(\mathbf{x}) = c(u_{\alpha_1}, u_{\alpha_2}) c(u_{\alpha_2}, u_{\alpha_3}) c(u_{\alpha_3}, u_{\alpha_4}) c(u_{\alpha_4}, u_{\alpha_5}) \prod_{k=1}^5 f(x_k)$$

(a)



$$f(\mathbf{x}) = c(u_{\beta_1}, u_{\beta_2}) c(u_{\beta_1}, u_{\beta_3}) c(u_{\beta_3}, u_{\beta_4}) c(u_{\beta_3}, u_{\beta_5}) \prod_{k=1}^5 f(x_k)$$

(b)

Figure 4.2: Two joint distributions with their associated copula function over five variables represented by (a) a chain graphical model and (b) a tree graphical model.

for proposing a new greedy algorithm for the MIMIC with copula functions. For instance, 1) choosing the two variables with the lowest copula entropy and set them the first elements in the chain, and 2) new chain links can be added to the chain by selecting variables with the lowest copula entropy according to the variables in the ends of the chain. This is shown in Algorithm 5.

Algorithm 5 Greedy algorithm to pick a permutation α in a chain model

- 1: Choose $(\alpha_m, \alpha_{m+1}) = \arg \min_{j \neq k} \widehat{H}(U_j, U_k)$, where $\widehat{H}()$ is an estimation of the copula entropy between the variables $u_j = F_{X_j}(x_j)$ and $u_k = F_{X_k}(x_k)$.
 - 2: Choose variables with the lowest copula entropy with respect to any of the ends of the chain. The constraint is to avoid a circular chain.
 - 3: The order of the chain defines permutation α .
-

For a tree graphical model is also possible to use the sum of pairwise mutual information for learning the optimal permutation β . This can be done by also using Kruskal's algorithm because the pairwise mutual information are replaced by the estimated copula entropies.

The relationship between the entropy of a bivariate copula function and the marginal mutual information of two variables, Equation (4.14), has both theoretical and practical importance: 1) the mutual information is given by the copula function *regardless* the marginal distributions, and 2) the estimation of the copula entropy can be more accurate than the estimation of mutual information because the copula domain is *always* bounded and standardized.

In this work the mutual information is estimated by using a Monte Carlo simulation. Algorithm 6 illustrates the sampling procedure for estimating the entropy of a bivariate copula function.

Algorithm 6 Monte Carlo method for estimating the copula entropy

- 1: Simulate several random samples $\{(u_i, v_i)\}_{i=1}^m$ from the copula distribution $c(u, v)$ with dependence parameter θ .
- 2: Calculate

$$\widehat{H}(U, V) = -\frac{1}{m} \sum_{i=1}^m \log c(u_i, v_i; \theta) .$$

The quantity $\widehat{H}(U, V)$ is an estimation of the copula entropy $H(U, V)$.

4.2.4 Sampling from Bayesian networks

Once the structure of a Bayesian network is known, a sample can be generated by following the order established by the conditional dependencies. The common choice for sampling from EDAs is the *Probabilistic Logic Sampling* (PLS) method (Henrion, 1988). In this algorithm, given an ancestral ordering $\pi = (\pi_1, \dots, \pi_d)$, a variable X_{π_i} is sampled after all its parents $\text{pa}(X_{\pi_i})$ have already been sampled. Algorithm 7 shows a pseudocode of the PLS.

Algorithm 7 Pseudocode of the PLS

```

1: Let  $\pi = (\pi_1, \dots, \pi_d)$  be an ancestral ordering of the variables.
2: for  $i = 1 \rightarrow N$  do
3:   for  $j = 1 \rightarrow d$  do
4:     Sample a value  $x_{\pi_j}$  for the variable  $X_{\pi_j}$  using the conditional distribu-
       tion  $f(x_{\pi_j} | \text{pa}(x_{\pi_j}))$ , where  $\text{pa}(x_{\pi_j})$  is the configuration already sam-
       pled for the parents of  $X_{\pi_j}$ .
5:   end for
6: end for

```

In order to use the PLS, variables must be ordered in such a way that the values for parents $\text{pa}(X_{\pi_i})$ must be assigned before X_{π_i} is sampled. An ordering of the variables satisfying such a property is called an *ancestral ordering*. For the graphical models presented in this section, the ancestral ordering is given when the permutation of indexes is found.

Some excellent references about PLS are (Jensen and Nielsen, 2007) and (Neapolitan, 2004). The reference (Larrañaga, 2002a) presents other methods for the simulation of Bayesian networks.

4.3 Our proposed EDAs

Our research has been conducted for designing new continuous EDAs. These EDAs are described below.

4.3.1 Incorporating copula functions

In (Salinas Gutiérrez et al., 2009; Salinas-Gutiérrez et al., 2009) we have presented an EDA based on a chain graphical model and bivariate copula functions. Every chain link in the graphical model represents the dependence between two decision variables. A bivariate copula function is used for modeling such dependence. The relationship between the mutual information and the bivariate copula entropy (Davy and Doucet, 2003) is used for measuring the dependence between variables. The probabilistic model used in this subsection can be seen as a generalization of the well known EDA MIMIC_c^G (Larrañaga et al., 1999, 2000a). In this work, for the first time, the estimation of copula parameters by means of the maximum likelihood function is presented.

Description of the Chain ^{Frank}_{Beta} EDA and the Chain ^{Gaussian}_{Beta} EDA

The EDAs presented in (Salinas-Gutiérrez et al., 2009) use a chain graphical model and two different dependence functions: a Frank copula and a Gaussian copula. These copulas are chosen because their dependence parameter have associated all range values of Kendall's tau. This means that negative and positive dependence between the marginals are considered in both copulas.

However, they differ in the way they model extreme and centered values (Trivedi and Zimmer, 2007). For instance, a Frank copula is mostly appropriate for data that exhibit weak dependence between extreme values and strong dependence between centered values.

In order to build the chain graphical model and according to Equation (4.14), the copula entropy between each pair of variables is calculated for estimating the mutual information. Then, the pair of variables with the largest mutual information are selected as the first chain link. The following variables (chain links) are chosen according to their mutual information with respect to the previous variable. Algorithm 8 shows a straightforward greedy algorithm for building the chain graphical model.

Algorithm 8 Greedy algorithm to pick a permutation α

- 1: Choose $(\alpha_1, \alpha_2) = \arg \min_{j \neq k} \widehat{H}(U_j, U_k)$, where $\widehat{H}()$ is an estimation of the copula entropy between the variables $u_j = F_{X_j}(x_j)$ and $u_k = F_{X_k}(x_k)$.
 - 2: Choose $\alpha_k = \arg \max_j \widehat{H}(U_{\alpha_{k-1}}, U_j)$, where $j \neq \alpha_1, \dots, \alpha_{k-1}$ and $k = 3, 4, \dots, d - 1, d$.
-

For a Gaussian copula there is a direct way to calculate its entropy and mutual information; for a Frank copula its entropy is estimated with a numerical approximation (Algorithm 6).

Once a permutation α is found, generating samples follows the steps of Algorithm 7. In order to do it, we first sample variable $U_{\alpha_1} \sim U(0, 1)$ and then we sample variables $U_{\alpha_k} \sim C(U_{\alpha_k} | U_{\alpha_{k-1}} = u_{\alpha_{k-1}})$ from the conditional copula of U_{α_k} given the value of $U_{\alpha_{k-1}}$ for $k = 2, \dots, d$. After that, the values of U_i are used to find quantiles X_i through expression $X_i = F_{X_i}^{-1}(U_i)$.

It is important to say that, by means of the copula theory, the MIMIC_c^G is a particular copula based EDA with Gaussian copulas and Gaussian marginal distributions.

For the proposed EDAs, Beta distributions are used as marginal distributions. In order to estimate the parameters of the probabilistic model, Equation (4.9), we use the Inference Function for Margins method (IFM) (Cherubini et al., 2004). This method is based on maximum likelihood and estimates first the parameters of marginal distributions and then use them to estimate parameters of copulas. The test problems used in this subsection have bounded search space. Each value of variable X_i from search space is transformed to a value in $(0, 1)$ through a linear transformation. This explains why we use Beta distributions as marginals.

We summarize in Algorithm 9 the proposed approach. The main aspects, such as the estimation of the probabilistic model and the generation of the new population, are also shown.

Algorithm 9 Pseudocode for estimating the model and generating a new population

```

1: for  $j = 1 \rightarrow d$  do
2:   For each variable  $X_j$ , estimate its marginal Beta parameters  $(a_j, b_j)$ .
3:   Determine  $U_j = F_j(X_j; a_j, b_j)$ , where  $F_j$  is the cumulative Beta distribution function.
4: end for
5: Calculate all pairwise concordance measures Kendall's tau.
6: Obtain an initial approximation to each bivariate dependence parameter using the corresponding relationship with Kendall's tau (Appendix A).
7: Estimate copula parameters using their initial approximations and the maximum likelihood method.
8: Calculate all pairwise copula entropies. For Frank copulas use Algorithm 6, for Gaussian copulas use Equation (5.12).
9: Use a greedy algorithm to pick a permutation  $\alpha$ , Algorithm 8.
10: Simulate  $U_{\alpha_1}$  from the uniform distribution  $U(0, 1)$ .
11: for  $k = 2 \rightarrow d$  do
12:   Simulate  $U_{\alpha_k}$  from conditional copula  $C(U_{\alpha_k} | U_{\alpha_{k-1}})$ .
13: end for
14: for  $j = 1 \rightarrow d$  do
15:   Determine  $X_j$  using quasi-inverse  $F_j^{-1}(U_j)$ .
16: end for

```

Experiments

We use three algorithms in order to optimize five test problems. One of these algorithms is MIMIC_c^G , the other two algorithms are the proposed EDAs. They are represented by the following notation:

- $\text{Chain}_{\text{Beta}}^{\text{Frank}}$: A chain graphical model with bivariate Frank copula functions and Beta marginal distributions.
- $\text{Chain}_{\text{Beta}}^{\text{Gaussian}}$: A chain graphical model with bivariate Gaussian copula functions and Beta marginal distributions.

The test problems used in the experiments are the Ackley, Griewangk, Rastrigin, Rosenbrock, and Sphere functions. These test functions are described in Appendix B. We use test problems in 10 dimensions. Each algorithm is run 30 times for each problem. The population size is 100. The maximum number of evaluations is 300,000. However, when convergence to a local minimum is detected the run is stopped. Any improvement less than 1×10^{-6} in 25 iterations is considered as convergence. The goal is to reach the optimum with an error less than 1×10^{-6} .

Numerical results

In Table 4.1 we report the fitness value reached by the algorithms in all test functions. The information about the number of evaluations required by each algorithm is reported in Table 4.2.

Table 4.1: Descriptive fitness results for all test functions.

Algorithm	Best	Median	Mean	Worst	Std. deviation
Ackley					
MIMIC _c ^G	6.47E-007	8.65E-007	8.62E-007	9.97E-007	1.06E-007
Chain _{Beta} ^{Frank}	5.79E-007	2.29E-006	3.06E-003	4.71E-002	9.31E-003
Chain _{Beta} ^{Gaussian}	5.62E-007	9.07E-007	3.64E-006	7.80E-005	1.41E-005
Griewangk					
MIMIC _c ^G	3.92E-007	8.66E-007	1.30E-003	3.88E-002	7.09E-003
Chain _{Beta} ^{Frank}	4.30E-007	9.38E-007	2.99E-003	2.90E-002	6.81E-003
Chain _{Beta} ^{Gaussian}	1.46E-007	8.11E-007	1.81E-002	4.31E-001	7.85E-002
Rastrigin					
MIMIC _c ^G	4.17E-007	9.96E-001	3.37E+000	2.33E+001	6.24E+000
Chain _{Beta} ^{Frank}	2.21E+000	4.99E+000	8.05E+000	3.69E+001	9.43E+000
Chain _{Beta} ^{Gaussian}	7.49E-007	4.00E+000	5.48E+000	2.68E+001	5.35E+000
Rosenbrock					
MIMIC _c ^G	7.31E+000	8.03E+000	8.89E+000	2.43E+001	3.17E+000
Chain _{Beta} ^{Frank}	6.87E+000	7.83E+000	7.95E+000	9.69E+000	6.44E-001
Chain _{Beta} ^{Gaussian}	6.26E+000	8.15E+000	8.53E+000	1.48E+001	1.78E+000
Sphere					
MIMIC _c ^G	3.55E-007	7.00E-007	7.10E-007	9.86E-007	2.02E-007
Chain _{Beta} ^{Frank}	3.39E-007	7.40E-007	3.03E-001	8.23E+000	1.50E+000
Chain _{Beta} ^{Gaussian}	3.42E-007	8.92E-007	4.85E-001	1.22E+001	2.23E+000

To properly compare the performance of the algorithms (using the optimum value reached), we conducted a hypothesis test for comparing the fitness averages. For all optimization problems, the test is applied to each pair of algorithms and is based on a Bootstrap method. Table 4.3 shows the confidence interval for the means, and the corresponding p-value.

Discussion

For the Ackley problem, intervals confidence show significant differences between MIMIC_c^G and Gaussian copula against Frank copula. This means that a dependence structure based on Gaussian copula is more adequate than a dependence structure based on Frank copula.

For the Griewangk problem, the algorithm that shows the best behaviour is MIMIC_c^G, closely followed by Frank copula algorithm. In this case, interval confidence between MIMIC_c^G and Gaussian copula shows that better results are obtained using both Gaussian dependence structure and marginals.

Table 4.2: Descriptive results for function evaluations in all test functions.

Algorithm	Mean	Std. deviation
Ackley		
MIMIC _c ^G	7660.30	131.24
Chain _{Beta} ^{Frank}	9310.30	1761.88
Chain _{Beta} ^{Gaussian}	7657.00	675.63
Griewangk		
MIMIC _c ^G	6927.70	1581.97
Chain _{Beta} ^{Frank}	8343.40	2460.13
Chain _{Beta} ^{Gaussian}	7835.20	2825.34
Rastrigin		
MIMIC _c ^G	11788.60	3146.69
Chain _{Beta} ^{Frank}	17055.40	5262.08
Chain _{Beta} ^{Gaussian}	15408.70	4511.01
Rosenbrock		
MIMIC _c ^G	12841.30	2665.61
Chain _{Beta} ^{Frank}	14280.10	1355.85
Chain _{Beta} ^{Gaussian}	14016.10	1666.29
Sphere		
MIMIC _c ^G	6175.30	154.87
Chain _{Beta} ^{Frank}	7069.60	2829.51
Chain _{Beta} ^{Gaussian}	7874.80	3144.74

The MIMIC_c^G is the algorithm that performed best for the Rastrigin problem. For this problem there is statistically significant difference in the mean fitness between the MIMIC_c^G and the Frank copula algorithms. Although results of Frank copula algorithm are not statistically different of Gaussian copula, is more suitable for this problem to choose a Gaussian structure than a Frank dependence. Respect to marginals distributions we can say something similar between Gaussian copula algorithm and MIMIC_c^G in the sense that is more adequate to choose Gaussian marginals than Beta marginals.

For the Rosenbrock problem, the intervals confidence shows statistical differences between MIMIC_c^G and Gaussian copula against Frank copula. In this case, a Frank dependence between marginals is more adequate than a Gaussian structure between marginals. Fitness results between MIMIC_c^G and Gaussian copula algorithm show no difference between Gaussian or Beta marginals if structure dependence is modeled by a Gaussian copula.

Finally, the fitness results for Sphere problem indicate that MIMIC_c^G obtained the global minimum in all the executions. The selection of Gaussian structure and Gaussian marginals is more adequate for this problem.

Regarding the number of fitness function evaluations, Table 4.2, the three algorithms performed in a similar way.

Table 4.3: Results for the difference between fitness means in each problem. A 95% interval confidence and a p-value are obtained through a Bootstrap technique.

Compared algorithms	95% Interval	p-value	
Ackley			
MIMIC _c ^G vs. Chain ^{Frank} _{Beta}	-6.15E-03	-7.37E-04	8.13E-02
MIMIC _c ^G vs. Chain ^{Gaussian} _{Beta}	-7.89E-06	1.69E-08	1.94E-01
Chain ^{Frank} _{Beta} vs. Chain ^{Gaussian} _{Beta}	7.28E-04	6.14E-03	8.17E-02
Griewangk			
MIMIC _c ^G vs. Chain ^{Frank} _{Beta}	-4.47E-03	1.29E-03	3.26E-01
MIMIC _c ^G vs. Chain ^{Gaussian} _{Beta}	-4.50E-02	-4.33E-04	1.62E-01
Chain ^{Frank} _{Beta} vs. Chain ^{Gaussian} _{Beta}	-4.34E-02	1.37E-03	2.60E-01
Rastrigin			
MIMIC _c ^G vs. Chain ^{Frank} _{Beta}	-8.11E+00	-1.48E+00	2.48E-02
MIMIC _c ^G vs. Chain ^{Gaussian} _{Beta}	-4.49E+00	3.20E-01	1.60E-01
Chain ^{Frank} _{Beta} vs. Chain ^{Gaussian} _{Beta}	-5.09E-01	5.87E+00	1.89E-01
Rosenbrock			
MIMIC _c ^G vs. Chain ^{Frank} _{Beta}	1.34E-01	2.01E+00	1.12E-01
MIMIC _c ^G vs. Chain ^{Gaussian} _{Beta}	-6.13E-01	1.50E+00	5.68E-01
Chain ^{Frank} _{Beta} vs. Chain ^{Gaussian} _{Beta}	-1.18E+00	-6.44E-02	9.48E-02
Sphere			
MIMIC _c ^G vs. Chain ^{Frank} _{Beta}	-8.51E-01	-1.16E-03	1.45E-01
MIMIC _c ^G vs. Chain ^{Gaussian} _{Beta}	-1.28E+00	-1.72E-02	1.42E-01
Chain ^{Frank} _{Beta} vs. Chain ^{Gaussian} _{Beta}	-9.84E-01	5.46E-01	6.80E-01

Conclusions

In this subsection we have introduced the use of bivariate copula functions for solving high dimensional optimization problems in EDAs. According to numerical experiments, the selection of the copula function for modeling the structure dependence and the selection of the marginal distribution can help achieving better fitness results. For each algorithm, we have fixed a copula family and a particular marginal distribution. However, it is not necessary to do it. We state that fitness results are a consequence of the selected copula functions and marginal distributions.

The three algorithms performed very similar, however, more experiments are necessary with different probabilistic models in order to identify where the copula functions mean a clear advantage to EDAs.

4.3.2 Incorporating a copula selection procedure

The previous works consider the use of a fixed copula function. Based on this observation, the works (Salinas-Gutiérrez et al., 2011b,a) are a recent contribution for selecting copula functions. A tree graphical model is employed in this

subsection. A copula function is selected for modeling the dependence between variables in each tree branch. The copula function is selected from a set of six bivariate copula functions and it is selected according to the likelihood function.

Description of the Chain $\overset{\text{Select}}{\text{Kernel}}$ EDA and the Tree $\overset{\text{Select}}{\text{Kernel}}$ EDA

As already described in section 4.2, it has been explained how to incorporate copula functions for modeling bivariate dependencies in tree branches and chain links. However, in order to gain modeling flexibility, it is not necessary that all chain links or tree branches are modeled by the same family of copula functions. In this thesis we propose a procedure for selecting copula functions in EDAs.

Two well known tools in statistics for model selection are the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). These criteria employ the maximized value of the likelihood function (\mathcal{L}_{\max}) for the estimated model, the number of parameters to be estimated (k) and the sample size (n):

$$\text{AIC} = -2 \ln(\mathcal{L}_{\max}) + 2k \quad (4.16)$$

$$\text{BIC} = -2 \ln(\mathcal{L}_{\max}) + k \ln(n) . \quad (4.17)$$

For a particular chain link or tree branch, the selection of a bivariate copula function from Appendix A can be done just by means of any of the criteria in Equations (4.16) and (4.17). However, in each generation of an EDA, the selected population has a fixed size. Moreover, the bivariate copula functions from Appendix A have one parameter. Hence, it can be noted that for any of the criteria AIC and BIC, the selection of a copula function just depends on the maximized value of the likelihood function, \mathcal{L}_{\max} . For that reason in this research work we propose a copula selection procedure based on the highest value of the log-likelihood function (Equation (3.22)).

In this thesis we assess the performance of two EDAs based on the chain and the tree graphical models. Both EDAs select the copula function for each chain link and tree branch. This modeling flexibility is not present in others EDAs based on copula functions, where just one copula function is previously chosen and used for modeling the dependence structure.

Experiments

In order to investigate the performance of the proposed EDAs, a benchmark of eleven functions for continuous optimization is utilized in the experiments. The conducted comparisons are made for contrasting the proposed algorithms with EDAs based on Gaussian distributions. Thus, we employ in the experiments four EDAs represented by the following notation

- Chain $\overset{\text{Gaussian}}{\text{Gaussian}}$: A chain graphical model with Gaussian copula functions and Gaussian marginals.

- Chain_{Kernel}^{Select} : A chain graphical model with the copula selection procedure and Gaussian kernels as marginals.
- Tree_{Gaussian}^{Gaussian} : A tree graphical model with Gaussian copula functions and Gaussian marginals.
- Tree_{Kernel}^{Select} : A tree graphical model with the copula selection procedure and Gaussian kernels as marginals.

Six copula functions are available to choose from Appendix A. These copula functions have appeared frequently in the literature and have been used in a broad range of applications. For instance, the Ali-Mikhail-Haq, Clayton, Farlie-Gumbel-Morgenstern, Frank and Gaussian copula functions can model negative and positive dependences between the marginals. One exception is the Gumbel copula which does not model negative dependence. The Ali-Mikhail-Haq and Farlie-Gumbel-Morgenstern copula functions are adequate for marginals with modest dependence. When dependence is strong between extremes values, the Clayton and Gumbel copula functions can model left and right tail association respectively. Frank copula is appropriate for data that exhibit weak dependence between extreme values and strong dependence between centered values, while Gaussian copula is adequate for data that exhibit weak dependence between centered values and strong dependence between extreme values.

Appendix B shows the description of the test functions to be minimized: Ackley, Cigar, Cigar Tablet, Ellipsoid, Griewangk, Rastrigin, Schwefel 1.2, Sphere Model, Trid, Two Axes, and Zakharov. The benchmark test suite includes separable functions and non-separable functions, from which there are unimodal and multimodal functions. In addition, the search domains are symmetric and asymmetric. All test functions are scalable.

We use test problems in 4, 8, 10, 12 and 16 dimensions. Each EDA is run 20 times for each problem. The population size is ten times the problem dimension. The maximum number of evaluations is 100,000. However, when convergence to a local minimum is detected the run is stopped. Any improvement less than 1×10^{-6} in 30 iterations is considered as premature convergence. The goal is to reach the optimum with an error less than 1×10^{-4} .

Numerical results

The results in dimensions 4, 8, and 16 for separable functions are reported in Table 4.4, whereas the results for non-separable functions are reported in Table 4.5. Tables 4.4 and 4.5 report descriptive statistics for the fitness values reached in all the runs. For each algorithmn and dimension, the minimum, median, mean, maximum, standard deviation and success rate are shown. The minimum (maximum) value reached is labelled best (worst). The success rate is the proportion of runs in which an algorithm found the global optimum.

4.3 Our proposed EDAs

Table 4.4: Descriptive results of the fitness for separable functions.

Algorithm	Dimension	Best	Median	Mean	Worst	Standard deviation	Success rate
Cigar							
Chain Gaussian	$d = 4$	2.05E-5	7.70E-5	2.08E-1	2.89E+0	6.65E-1	0.70
	$d = 8$	2.21E-5	8.32E-5	3.00E-1	4.36E+0	9.91E-1	0.80
	$d = 16$	3.56E-5	6.95E-5	1.04E-2	2.01E-1	4.49E-2	0.90
Chain Select Kernel	$d = 4$	1.47E-5	8.00E-5	3.49E+0	6.98E+1	1.56E+1	0.80 +
	$d = 8$	3.85E-5	7.14E-5	7.00E-5 *	9.98E-5	1.76E-5	1.00 +
	$d = 16$	3.70E-5	7.95E-5	7.58E-5	9.88E-5	1.72E-5	1.00 +
Tree Gaussian	$d = 4$	1.26E-5	7.61E-5	1.13E+0	8.64E+0	2.70E+0	0.75
	$d = 8$	3.19E-5	7.80E-5	1.88E-2	2.70E-1	6.24E-2	0.80
	$d = 16$	3.22E-5	9.03E-5	2.71E-1	3.33E+0	7.63E-1	0.70
Tree Select Kernel	$d = 4$	7.20E-6	8.55E-5	1.60E+3	3.18E+4	7.10E+3	0.65
	$d = 8$	1.79E-5	7.42E-5	8.99E-4 *	1.66E-2	3.70E-3	0.95 +
	$d = 16$	4.74E-5	7.85E-5	7.74E-5 *	9.57E-5	1.43E-5	1.00 +
Cigar Tablet							
Chain Gaussian	$d = 4$	1.50E-5	7.04E-5	4.43E+2	8.84E+3	1.98E+3	0.70
	$d = 8$	3.26E-5	8.18E-5	4.86E-1	5.31E+0	1.39E+0	0.80
	$d = 16$	2.87E-5	7.15E-5	6.92E-5	9.99E-5	1.86E-5	1.00
Chain Select Kernel	$d = 4$	1.81E-5	6.67E-5	3.49E+1	6.25E+2	1.40E+2	0.70
	$d = 8$	4.51E-5	7.72E-5	7.50E-5 *	9.95E-5	1.92E-5	1.00 +
	$d = 16$	4.47E-5	8.10E-5	7.86E-5	9.65E-5	1.69E-5	1.00
Tree Gaussian	$d = 4$	6.93E-6	9.40E-5	3.85E+0	5.08E+1	1.16E+1	0.55
	$d = 8$	3.28E-5	7.09E-5	7.45E+2	1.49E+4	3.33E+3	0.95
	$d = 16$	4.63E-5	8.38E-5	5.12E-1	1.02E+1	2.29E+0	0.90
Tree Select Kernel	$d = 4$	1.55E-5	9.27E-5	1.65E+0	1.38E+1	3.86E+0	0.55
	$d = 8$	3.82E-5	7.74E-5	6.98E-5	9.01E-5	1.70E-5	1.00 +
	$d = 16$	5.94E-5	8.49E-5	8.14E-5	9.97E-5	1.36E-5	1.00 +
Ellipsoid							
Chain Gaussian	$d = 4$	2.19E-5	6.72E-5	9.78E+0	1.95E+2	4.36E+1	0.85
	$d = 8$	2.49E-5	6.26E-5	3.24E+0	6.34E+1	1.42E+1	0.70
	$d = 16$	5.20E-5	8.49E-5	2.26E+1	2.57E+2	6.83E+1	0.75
Chain Select Kernel	$d = 4$	2.28E-5	8.41E-5	5.43E-1	8.40E+0	1.90E+0	0.75
	$d = 8$	2.30E-5	7.00E-5	6.65E-5 *	9.53E-5	1.96E-5	1.00 +
	$d = 16$	4.87E-5	8.34E-5	7.95E-5 *	9.78E-5	1.59E-5	1.00 +
Tree Gaussian	$d = 4$	1.44E-5	7.82E-5	4.25E+1	8.41E+2	1.88E+2	0.65
	$d = 8$	4.54E-5	7.69E-5	1.08E-1	1.21E+0	3.29E-1	0.80
	$d = 16$	3.88E-5	9.33E-5	4.29E-1	3.33E+0	9.40E-1	0.55
Tree Select Kernel	$d = 4$	1.26E-5	6.79E-5	6.19E-2 *	6.76E-1	1.85E-1	0.80 +
	$d = 8$	3.86E-5	7.68E-5	7.51E-5 **	9.76E-5	1.66E-5	1.00 +
	$d = 16$	4.32E-5	8.27E-5	7.93E-5 **	9.95E-5	1.62E-5	1.00 +
Rastrigin							
Chain Gaussian	$d = 4$	3.83E-5	2.39E+0	2.31E+0	4.73E+0	1.56E+0	0.10
	$d = 8$	1.10E+1	2.04E+1	2.02E+1	2.77E+1	4.40E+0	0.00
	$d = 16$	5.74E+1	7.40E+1	7.41E+1	8.64E+1	8.65E+0	0.00
Chain Select Kernel	$d = 4$	3.20E-6	7.82E-5	9.75E-1 **	8.55E+0	2.28E+0	0.75 +
	$d = 8$	3.97E-5	1.55E+1	1.38E+1 **	2.89E+1	1.07E+1	0.15 +
	$d = 16$	5.18E+1	7.42E+1	7.52E+1	9.39E+1	1.14E+1	0.00
Tree Gaussian	$d = 4$	1.25E-5	2.31E+0	2.55E+0	5.17E+0	1.49E+0	0.10
	$d = 8$	1.03E+1	1.96E+1	1.87E+1	2.59E+1	4.45E+0	0.00
	$d = 16$	6.16E+1	7.28E+1	7.29E+1	8.55E+1	6.57E+0	0.00
Tree Select Kernel	$d = 4$	5.48E-7	8.62E-5	6.40E-1 **	5.80E+0	1.35E+0	0.60 +
	$d = 8$	3.42E-5	1.32E+1	1.05E+1 **	2.92E+1	1.04E+1	0.40 +
	$d = 16$	2.66E+1	8.22E+1	7.89E+1	1.04E+2	1.57E+1	0.00
Sphere Model							
Chain Gaussian	$d = 4$	5.90E-6	6.81E-5	1.37E-2	2.74E-1	6.11E-2	0.90
	$d = 8$	3.59E-5	7.39E-5	7.22E-5	9.99E-5	1.84E-5	1.00
	$d = 16$	4.25E-5	7.75E-5	7.62E-5	9.77E-5	1.53E-5	1.00
Chain Select Kernel	$d = 4$	2.54E-5	5.10E-5	5.43E-5 *	9.99E-5	2.16E-5	1.00 +
	$d = 8$	2.16E-5	7.84E-5	7.07E-5	9.65E-5	2.35E-5	1.00
	$d = 16$	5.29E-5	8.22E-5	8.00E-5	9.95E-5	1.29E-5	1.00
Tree Gaussian	$d = 4$	3.90E-6	4.38E-5	5.08E-5	9.86E-5	3.08E-5	1.00
	$d = 8$	3.19E-5	7.05E-5	6.74E-5	9.77E-5	1.75E-5	1.00
	$d = 16$	3.92E-5	7.09E-5	7.06E-5	9.57E-5	1.41E-5	1.00
Tree Select Kernel	$d = 4$	8.63E-6	6.40E-5	9.46E-3	1.12E-1	2.78E-2	0.80
	$d = 8$	1.50E-5	5.64E-5	5.97E-5	9.75E-5	2.19E-5	1.00
	$d = 16$	4.89E-5	8.23E-5	8.10E-5	9.70E-5	1.24E-5	1.00
Two Axes							
Chain Gaussian	$d = 4$	9.16E-6	6.35E-5	2.73E+0	4.20E+1	9.50E+0	0.75
	$d = 8$	2.49E-5	9.49E-5	7.83E-1	4.18E+0	1.30E+0	0.60
	$d = 16$	4.86E-5	9.60E-5	4.16E-1	4.91E+0	1.14E+0	0.55
Chain Select Kernel	$d = 4$	2.89E-5	2.88E-3	8.60E-1	1.08E+1	2.59E+0	0.50
	$d = 8$	3.48E-5	6.88E-5	6.62E-5 **	9.06E-5	1.52E-5	1.00 +
	$d = 16$	4.14E-5	8.05E-5	7.81E-5 *	9.79E-5	1.64E-5	1.00 +
Tree Gaussian	$d = 4$	2.08E-5	7.61E-5	6.14E-1	1.03E+1	2.31E+0	0.70
	$d = 8$	2.15E-5	1.27E-2	1.51E+0	1.19E+1	2.89E+0	0.40
	$d = 16$	5.30E-5	1.87E-3	5.24E-1	3.80E+0	9.78E-1	0.50
Tree Select Kernel	$d = 4$	1.79E-5	8.71E-5	1.02E-1	1.69E+0	3.76E-1	0.65
	$d = 8$	3.18E-5	7.15E-5	6.93E-5 **	9.73E-5	1.42E-5	1.00 +
	$d = 16$	5.83E-5	7.99E-5	7.87E-5 **	9.78E-5	1.17E-5	1.00 +

* denotes that the EDA with copula selection procedure outperforms the corresponding EDA with Gaussian copula, at $\alpha = 0.10$
 ** denotes that the EDA with copula selection procedure outperforms the corresponding EDA with Gaussian copula, at $\alpha = 0.05$
 + denotes that the EDA with copula selection procedure has greater success rate than the EDA with Gaussian copula

4. EDAs and Copula Functions

Table 4.5: Descriptive results of the fitness for non-separable functions.

Algorithm	Dimension	Best	Median	Mean	Worst	Standard deviation	Success rate
Ackley							
Chain Gaussian	$d = 4$	2.15E-5	7.34E-5	9.11E-3	1.65E-1	3.68E-2	0.90
	$d = 8$	5.56E-5	8.48E-5	8.11E-5	9.94E-5	1.51E-5	1.00
	$d = 16$	7.27E-5	8.85E-5	8.82E-5	9.94E-5	7.43E-6	1.00
Chain Select Kernel	$d = 4$	2.80E-5	7.04E-5	6.99E-5 *	9.74E-5	2.12E-5	1.00 +
	$d = 8$	6.40E-5	8.92E-5	8.49E-5	9.87E-5	1.15E-5	1.00
	$d = 16$	7.59E-5	9.20E-5	9.00E-5	9.88E-5	7.27E-6	1.00
Tree Gaussian	$d = 4$	3.72E-5	6.25E-5	6.71E-5	1.49E-4	2.63E-5	0.90
	$d = 8$	5.03E-5	7.88E-5	7.91E-5	9.88E-5	1.52E-5	1.00
	$d = 16$	7.29E-5	9.24E-5	8.85E-5	9.92E-5	8.80E-6	1.00
Tree Select Kernel	$d = 4$	2.57E-5	7.16E-5	1.19E-2	1.94E-1	4.36E-2	0.85
	$d = 8$	4.81E-5	8.72E-5	8.39E-5	9.82E-5	1.24E-5	1.00
	$d = 16$	6.13E-5	9.13E-5	8.82E-5	9.99E-5	1.04E-5	1.00
Griewangk							
Chain Gaussian	$d = 4$	3.43E-2	1.01E-1	1.10E-1	2.20E-1	5.28E-2	0.00
	$d = 8$	1.81E-1	3.67E-1	3.69E-1	5.53E-1	9.73E-2	0.00
	$d = 16$	3.58E-5	8.18E-5	7.81E-5	9.92E-5	1.79E-5	1.00
Chain Select Kernel	$d = 4$	2.96E-2	1.02E-1	1.19E-1	2.83E-1	6.68E-2	0.00
	$d = 8$	7.40E-3	2.03E-1	2.21E-1 **	5.22E-1	1.31E-1	0.00
	$d = 16$	6.51E-5	8.02E-5	8.15E-5	9.92E-5	9.65E-6	1.00
Tree Gaussian	$d = 4$	4.79E-2	1.26E-1	1.26E-1	2.21E-1	5.21E-2	0.00
	$d = 8$	1.58E-1	3.80E-1	3.78E-1	5.49E-1	1.10E-1	0.00
	$d = 16$	4.41E-5	7.60E-5	7.68E-5	9.70E-5	1.30E-5	1.00
Tree Select Kernel	$d = 4$	2.90E-2	8.92E-2	1.02E-1 *	2.06E-1	5.05E-2	0.00
	$d = 8$	5.40E-5	2.45E-1	2.53E-1 **	5.34E-1	1.52E-1	0.05 +
	$d = 16$	2.93E-5	7.59E-5	7.50E-5	9.97E-5	1.80E-5	1.00
Schwefel 1.2							
Chain Gaussian	$d = 4$	2.53E-5	6.00E-3	1.49E-1	9.44E-1	2.76E-1	0.40
	$d = 8$	9.94E-5	4.89E-2	2.82E-1	2.04E+0	5.92E-1	0.10
	$d = 16$	1.38E-2	3.94E-1	1.07E+0	5.60E+0	1.56E+0	0.00
Chain Select Kernel	$d = 4$	1.84E-5	9.74E-5	1.43E-2 **	1.41E-1	3.58E-2	0.60 +
	$d = 8$	4.95E-5	1.01E-4	8.38E-3 **	7.25E-2	1.83E-2	0.50 +
	$d = 16$	9.54E-5	2.77E-4	2.14E-2 **	2.21E-1	5.08E-2	0.40 +
Tree Gaussian	$d = 4$	5.82E-6	7.78E-5	1.02E-2	5.44E-2	1.78E-2	0.55
	$d = 8$	9.65E-5	1.94E-2	4.50E-1	3.44E+0	8.87E-1	0.05
	$d = 16$	3.66E-2	4.34E-1	1.15E+0	6.78E+0	1.81E+0	0.00
Tree Select Kernel	$d = 4$	3.03E-5	8.15E-5	7.26E-2	1.39E+0	3.10E-1	0.65 +
	$d = 8$	5.21E-5	9.52E-5	3.84E-4 **	2.66E-3	7.50E-4	0.75 +
	$d = 16$	9.47E-5	2.89E-4	4.85E-3 **	6.74E-2	1.54E-2	0.40 +
Trid							
Chain Gaussian	$d = 4$	-1.60E+1	-1.60E+1	-1.58E+1	-1.17E+1	9.63E-1	0.65
	$d = 8$	-1.12E+2	-1.12E+2	-1.11E+2	-1.04E+2	2.14E+0	0.65
	$d = 16$	-8.00E+2	-8.00E+2	-7.92E+2	-6.92E+2	2.41E+1	0.55
Chain Select Kernel	$d = 4$	-1.60E+1	-1.60E+1	-1.60E+1	-1.56E+1	9.51E-2	0.80 +
	$d = 8$	-1.12E+2	-1.12E+2	-1.12E+2 *	-1.12E+2	7.85E-4	0.90 +
	$d = 16$	-8.00E+2	-8.00E+2	-8.00E+2 *	-7.99E+2	1.26E-1	0.60 +
Tree Gaussian	$d = 4$	-1.60E+1	-1.60E+1	-1.59E+1	-1.48E+1	2.68E-1	0.65
	$d = 8$	-1.12E+2	-1.12E+2	-1.12E+2	-1.09E+2	8.04E-1	0.55
	$d = 16$	-8.00E+2	-8.00E+2	-7.99E+2	-7.93E+2	1.68E+0	0.45
Tree Select Kernel	$d = 4$	-1.60E+1	-1.60E+1	-1.60E+1 *	-1.60E+1	1.86E-4	0.90 +
	$d = 8$	-1.12E+2	-1.12E+2	-1.12E+2	-1.10E+2	6.22E-1	0.65 +
	$d = 16$	-8.00E+2	-8.00E+2	-8.00E+2	-7.97E+2	9.42E-1	0.60 +
Zakharov							
Chain Gaussian	$d = 4$	3.84E-5	9.59E-5	1.26E-1	2.13E+0	4.78E-1	0.55
	$d = 8$	8.76E-5	1.36E-2	1.23E-1	9.53E-1	2.38E-1	0.05
	$d = 16$	8.52E-5	2.56E-3	7.51E-3	3.13E-2	9.44E-3	0.20
Chain Select Kernel	$d = 4$	1.05E-5	6.76E-5	4.47E-3 *	8.44E-2	1.88E-2	0.90 +
	$d = 8$	4.18E-5	9.80E-5	2.51E-3 **	4.62E-2	1.03E-2	0.80 +
	$d = 16$	5.93E-5	9.79E-5	1.65E+1	1.45E+2	3.66E+1	0.70 +
Tree Gaussian	$d = 4$	1.99E-5	7.41E-4	1.87E-2	1.35E-1	3.90E-2	0.50
	$d = 8$	9.92E-5	8.63E-3	6.46E-2	4.05E-1	1.13E-1	0.05
	$d = 16$	8.52E-5	3.34E-2	3.56E-1	4.80E+0	1.06E+0	0.15
Tree Select Kernel	$d = 4$	1.58E-5	7.83E-5	2.10E-1	2.60E+0	6.55E-1	0.65 +
	$d = 8$	3.86E-5	9.87E-5	1.59E-2 *	3.11E-1	6.94E-2	0.85 +
	$d = 16$	8.07E-5	9.05E-5	1.40E+1	1.22E+2	3.65E+1	0.80 +

* denotes that the EDA with copula selection procedure outperforms the corresponding EDA with Gaussian copula, at $\alpha = 0.10$
 ** denotes that the EDA with copula selection procedure outperforms the corresponding EDA with Gaussian copula, at $\alpha = 0.05$
 † denotes that the EDA with copula selection procedure has greater success rate than the EDA with Gaussian copula

In order to properly compare the performance of the algorithms, we conducted a hypothesis test to determine if the copula selection procedure can

help in improving results. The hypotheses for the test are $H_0 : \mu_a \leq \mu_b$ vs. $H_1 : \mu_a > \mu_b$, where μ_a stands for the fitness average of an EDA based on Gaussian copula and μ_b stands for the fitness average of an EDA based on copula selection. The statistical comparisons are for the algorithms with the same graphical model. The hypothesis test is based on a Bootstrap method. When a null hypothesis can be rejected, the corresponding average μ_b is marked with an asterisk (*).

Besides the average fitness, the success rate is another performance measure that can help in the comparisons of the algorithms. Figures 4.3 and 4.4 show respectively the success rate in each dimension for separable and non-separable functions. If the success rate of an EDA based on copula selection is greater than the success rate of the corresponding algorithm without copula selection, it is marked with a plus sign (+).

Discussion

Separable functions. In general, the unimodal and separable functions are not difficult to solve. The algorithms have a good performance in solving all the functions with exception of the multimodal Rastrigin function. However, the statistical tests and the success rates show that the EDAs based on copula selection can achieve and outperform the results founded by the algorithms based on Gaussian copula.

The success rates for EDAs based on copula selection do not decrease with the dimension problem in all separable functions, except the Rastrigin function. Moreover, their success rate in functions Cigar, Cigar Tablet, Ellipsoid, Sphere model, and Two Axes is always 100% in dimension 16.

Non-separable functions. It is known that this kind of test functions have associations or dependencies among their variables, thus, they are difficult to solve. The performance of all algorithms in the Ackley and Griewangk functions shows that their success rate does not decrease its value with the dimension problem. For the Schwefel 1.2, Trid, and Zakharov functions, all the algorithms success rates decrease with the dimensionality.

Conclusions

In this subsection a copula selection procedure for continuous EDAs has been introduced. The implementation of this procedure is shown in two well known graphical models. According to the numerical experiments the selection of a copula function for modeling the dependence structure can help achieving better fitness values. This means that dependencies between decision variables must be modeled adequately in order to get good solutions.

The EDA based on a chain graphical model and the EDA based on a tree graphical model, both with a copula selection procedure, have a better performance than EDAs based on multivariate Gaussian distributions. The success rate already indicates a better performance of the algorithms adapted with cop-

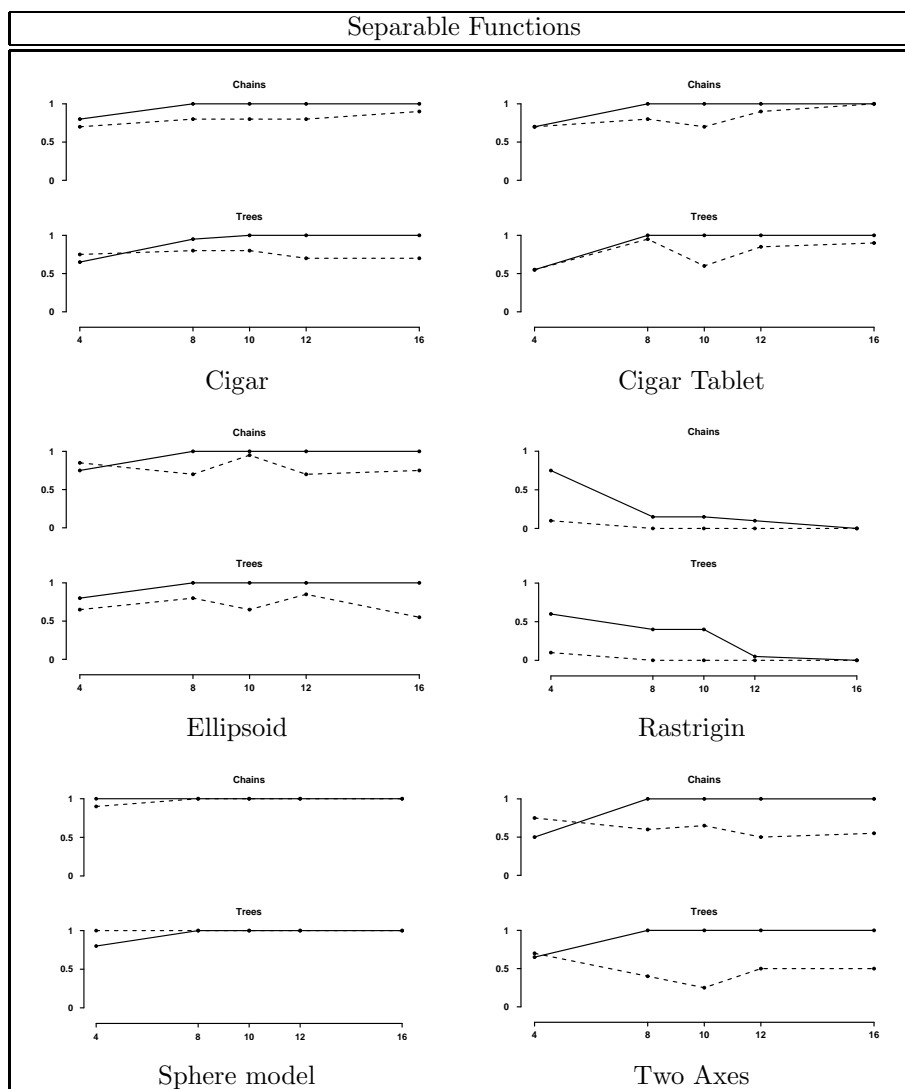


Figure 4.3: Success rate in each dimension for separable functions. The horizontal axis represents the dimension problem and the vertical axis represents the success rate. The solid line is used for the EDAs based on copula selection and the dashed line for the EDAs based on the Gaussian copula.

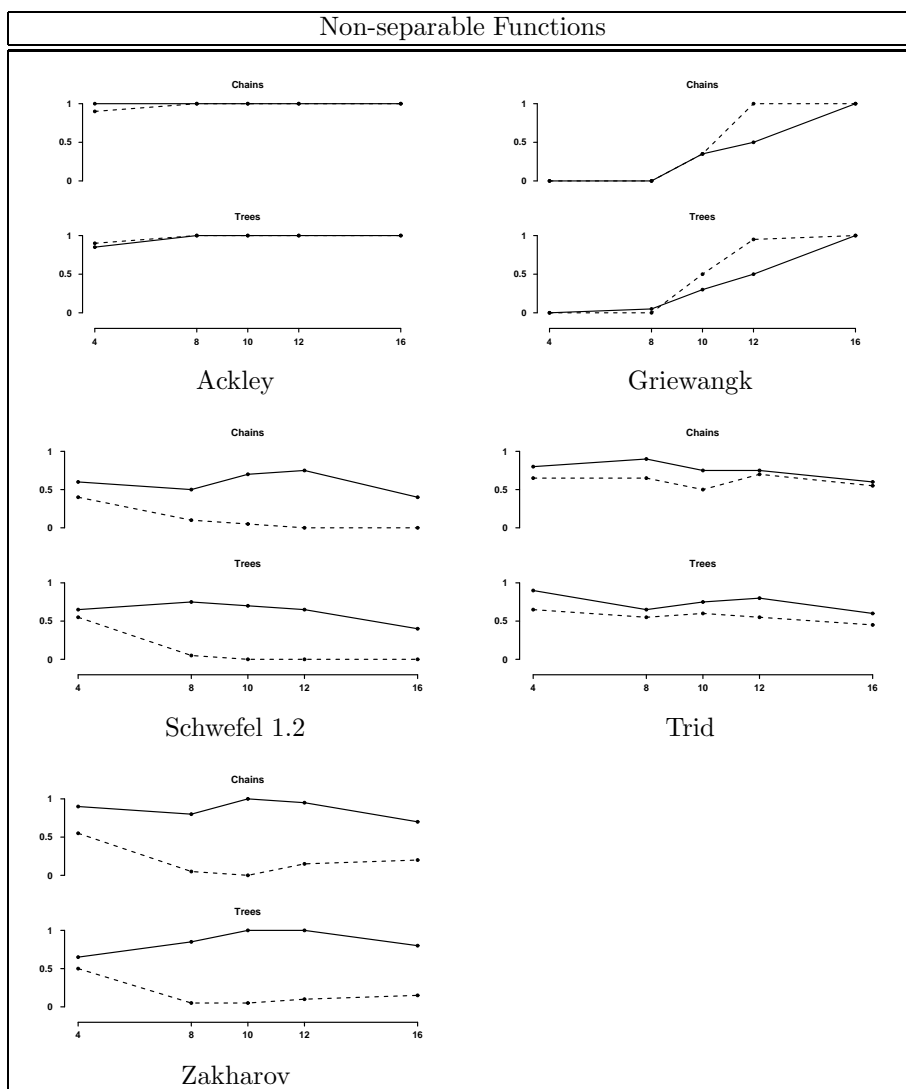


Figure 4.4: Success rate in each dimension for non-separable functions. The horizontal axis represents the dimension problem and the vertical axis represents the success rate. The solid line is used for the EDAs based on copula selection and the dashed line for the EDAs based on the Gaussian copula.

ula selection in higher dimension. Although the statistical comparisons show similar performances of both algorithms in reaching the function optima, the success rate clearly indicates that EDAs with copula selection are more robust since they reach the optima in a larger number of experiments. For every function six comparisons are shown in the Tables 4.4 and 4.5, that is, two algorithms times three dimensions. The tables show a total of 66 comparisons. Out of the 36 comparisons for the separable functions, the EDA with copula selection excels in 22 cases, it has similar performance in 9 cases, and it is outperformed in only 5 cases. Similarly, out of the 30 comparisons for the non separable functions, the EDA with copula selection excels in 20 cases, it has similar performance in 9 cases, and it is outperformed in only one case. This could imply that the EDAs based on a copula selection procedure are more adequate than algorithms with a fixed dependence function for non-separable optimization problems. Nonetheless, more experiments are necessary with different graphical models in order to identify where the copula functions have a clear advantage to EDAs.

4.4 Summary

We have presented in this chapter the incorporation of copula functions into continuous EDAs. Moreover, we have also shown how the structure of a probabilistic model can be learnt by taking into account the dependence among variables, regardless the behavior of marginal distributions. The proposed EDAs presented in this chapter use the copula entropy as a measure of dependence.

The EDAs presented in this chapter integrate copula functions into graphical models. From a theoretical point of view, this provides the following advantages: (1) the most important dependencies are represented by the graphical model, (2) dependencies can be linear or nonlinear, (3) any joint distribution can be factorized by copula functions of lower order, (4) copula functions can be from different family, and (5) marginal distributions can be selected separately.

Chapter 5

The D-vine EDA

The goal of this chapter is to present regular vines and show how they can be incorporated into continuous EDAs. Regular vines are graphical models that represent multivariate distributions using bivariate and conditional bivariate copula functions. In particular, a subset of regular vines known as D-vine is adapted for optimizing several benchmark functions.

5.1 Regular vines

A class of undirected graphs for representing high dimensional probability distributions are named vines. These kind of graphs use bivariate and conditional bivariate copula functions.

According to Kurowicka and Cooke (2002), a vine on d variables is a set of nested trees, where the edges of the tree j are the nodes of the tree $j + 1$, for $j = 1, \dots, d - 2$, and each tree has the maximum number of edges. We illustrate the concept of a vine in the following example.

Example 5.1 (Three dimensions). Let (X_1, X_2, X_3) be a three dimensional random vector with a joint density function $f(x_1, x_2, x_3)$. A well known factorization for the trivariate density is given by the expression

$$f(x_1, x_2, x_3) = f(x_1) \cdot f(x_2|x_1) \cdot f(x_3|x_1, x_2) . \quad (5.1)$$

From the copula theory and Equation 3.2, the joint density can also be factorized as

$$f(x_1, x_2, x_3) = f(x_1) \cdot f(x_2) \cdot f(x_3) \cdot c(u_1, u_2, u_3) . \quad (5.2)$$

However, once again by means of Equation (3.2), we can decompose the conditional distributions into bivariate copulas and marginal densities

$$f(x_2|x_1) = \frac{f(x_1, x_2)}{f(x_1)} = c(u_1, u_2) \cdot f(x_2) , \quad (5.3)$$

$$\begin{aligned}
 f(x_3|x_1, x_2) &= \frac{f(x_1, x_2, x_3)}{f(x_1, x_2)} \\
 &= \frac{f(x_1, x_3|x_2) \cdot f(x_2)}{f(x_1|x_2) \cdot f(x_2)} \\
 &= \frac{c(u_1, u_3|u_2) \cdot f(x_1|x_2) \cdot f(x_3|x_2)}{f(x_1|x_2)} \\
 &= c(u_1, u_3|u_2) \cdot c(u_2, u_3) \cdot f(x_3) . \tag{5.4}
 \end{aligned}$$

Inserting expressions (5.3) and (5.4) into Equation (5.1) gives

$$f(\mathbf{x}) = c(u_1, u_2) \cdot c(u_2, u_3) \cdot c(u_1, u_3|u_2) \cdot \prod_{k=1}^3 f(x_k) . \tag{5.5}$$

The pair copula decomposition in Equation (5.5) can be represented by a graphical structure. Figure 5.1 (a) shows a graph with 6 nodes and 6 edges. Figure 5.1 (b) shows a vine with 5 nodes, 2 trees and 3 edges. The contents of each node in the vine represents the indexes of the random variables. For example, the node with number two is related to random variables (X_2, U_2) . Two edges in the vine are associated to marginal bivariate copulas, whereas one edge is associated to a conditional bivariate copula.

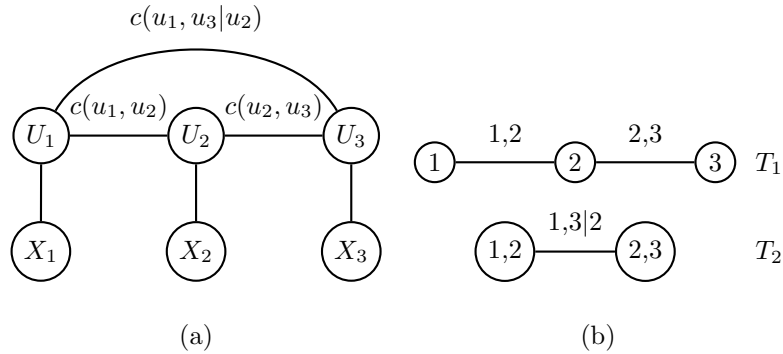


Figure 5.1: (a) An undirected graphical model. (b) A typical vine representation. Both graphs, (a) and (b), refer to the trivariate density function (5.5). See text for details.

By comparing Equations (5.2) and (5.5), we see that a trivariate copula density can be built using only bivariate copulas as building blocks:

$$c(u_1, u_2, u_3) = \underbrace{c(u_1, u_2)}_{\text{marginals}} \cdot \underbrace{c(u_1, u_3|u_2)}_{\text{conditional}} . \tag{5.6}$$

From Figure 5.1 (b), it can be noticed that the edges of the first tree T_1 are the nodes of the second tree T_2 and each tree has the maximum number of edges, i.e., two and one edges respectively. Moreover, from Equation 5.6, it can be seen that the tree T_1 is related to the marginal bivariate copulas and the nested tree T_2 is related to the conditional bivariate copula. ◀

Several comments can be said from the exposition of Example 5.1. For example, the vine representation for the joint density function is not unique. There are six different permutations for the indexes of variables, but only three permutations give different factorizations.

Vines give a way of extending bivariate copula functions to higher dimensions. By selecting an adequate set of bivariate copula functions, it is possible to design new d -dimensional copulas. Moreover, vines can be easily adapted to higher dimensions.

Finally, besides vines are graphical representations of pair copula decompositions, they can provide a more flexible representation of the joint distribution.

In this thesis, we are interested in using a special subset of vines as probabilistic models in EDAs. We refer to *regular vines* and present its formal definition (Kurowicka and Cooke, 2006).

Definition 5.1 (Regular vine). \mathcal{V} is a regular vine on d elements if

1. $\mathcal{V} = (T_1, \dots, T_{d-1})$, where T_i is a tree¹ for all $i = 1, \dots, d - 1$.
2. T_1 is a connected tree with nodes $N_1 = \{1, \dots, d\}$ and edges E_1 . For $i = 2, \dots, d - 1$, $T_i = (N_i, E_i)$ is a connected tree with nodes $N_i = E_{i-1}$.
3. For $i = 2, \dots, d - 1$, if $\{a, b\} \in E_i$, then $\#a \Delta b = 2$, where Δ denotes the symmetric difference. In other words, if a and b are nodes of T_i connected by an edge in T_i , where $a = \{a_1, a_2\}$ and $b = \{b_1, b_2\}$, then exactly one of the a_i equals one of the b_i . This condition is called the *proximity condition*.

The first and second properties in Definition 5.1 refer to vines. Third property (Bedford and Cooke, 2001, 2002) refers to the proximity condition in a regular vine, since it expresses the fact that two edges in tree j are joined by an edge in tree $j + 1$ only if these edges share a common node, $j = 1, \dots, d - 2$.

Two families of regular vines are the *D-vine* and the *canonical vine (C-vine)*². These special cases of regular vines impose additional restrictions and are characterized by minimal and maximal degrees of nodes in the trees.

Definition 5.2 (C-vine, D-vine). A regular vine is called a

1. **Canonical** or **C-vine** if each tree T_i has a unique node of degree $d - i$. The node with maximal degree in T_1 is the *root*.

¹A tree, as defined in Kurowicka and Cooke (2006), can be considered as a forest of trees. A tree in which all nodes are connected is termed as a *connected tree*

²D-vines were originally called *drawable vines*, while canonical vines owe their name to the fact that they are the most natural for sampling (Kurowicka and Cooke, 2006).

2. **D-vine** if each node in T_1 has a degree of at most 2.

Examples of canonical and D-vines on 4 nodes are shown in Figures 5.2 and 5.3 respectively.

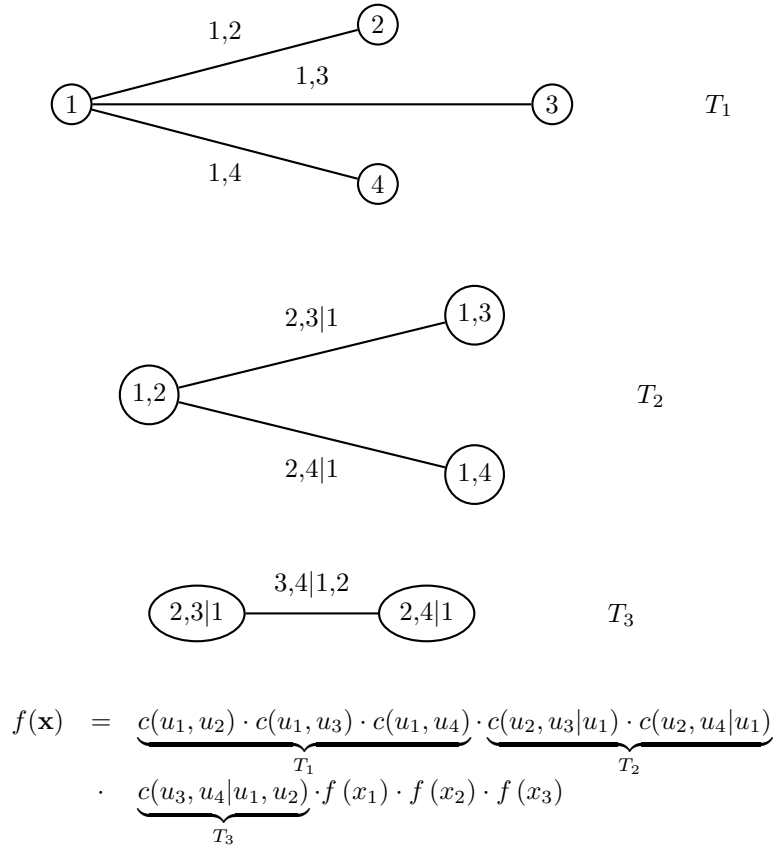
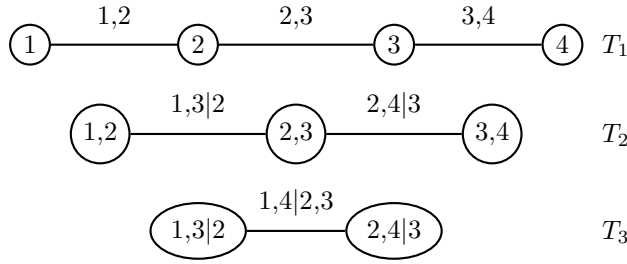


Figure 5.2: Example of a four-dimensional C-vine.

Figure 5.3 shows a D-vine on four variables. The tree T_1 is built on marginal pairwise variables, whereas the tree T_2 takes into account conditional pairwise variables. Observe how tree T_2 is built on tree T_1 . The last tree, T_3 , involves variables U_1 and U_4 conditioned on variables U_2 , and U_3 . Every tree, except tree T_1 , has associated conditional bivariate distributions. References (Aas et al., 2009; Kurowicka and Cooke, 2006) provide formal definitions of regular vines and illustrative information.

From a theoretical point of view, it is possible to model any d -dimensional dependence structure by means of a D-vine and bivariate copulas. However, for practical purposes, it is not necessary to build the complete D-vine, it is enough to select an adequate permutation of variables in order to define tree T_1 with



$$\begin{aligned}
 f(\mathbf{x}) &= \underbrace{c(u_1, u_2) \cdot c(u_2, u_3) \cdot c(u_3, u_4)}_{T_1} \cdot \underbrace{c(u_1, u_3|u_2) \cdot c(u_2, u_4|u_3)}_{T_2} \\
 &\cdot \underbrace{c(u_1, u_4|u_2, u_3)}_{T_3} \cdot f(x_1) \cdot f(x_2) \cdot f(x_3)
 \end{aligned}$$

Figure 5.3: Example of a four-dimensional D-vine.

the greatest information about the d -dimensional distribution. After the tree T_1 is chosen, we can build tree T_2 to increase the information of tree T_1 .

Before presenting the implementation of a D-vine into an EDA, we provide some theoretical relationships between multivariate distributions and their associated copula functions.

5.2 Copulas and information theory measures

In this thesis, the Kullback-Leibler divergence has been used as a measure of the difference between two probability distributions. Below, we prove an important relationship between the Kullback-Leibler divergence and copula functions.

Proposition 5.1. *Let f and g be two d -dimensional density functions with marginal densities f_i and g_i , respectively for $i = 1, \dots, d$. Then, the Kullback-Leibler divergence between multivariate densities f and g is given by the expression,*

$$D_{KL}(f||g) = \sum_{i=1}^d D_{KL}(f_i||g_i) + D_{KL}(c_f||c_g) \ ,$$

where c_f and c_g are the associated copula functions for multivariate densities f and g .

Proof. By definition, we have that

$$\begin{aligned}
 D_{KL}(f||g) &= E_{f(\mathbf{x})} \left[\log \frac{f(\mathbf{x})}{g(\mathbf{x})} \right] \\
 &= E_{f(\mathbf{x})} \left[\log \frac{\prod_{i=1}^d f_i \cdot c_f}{\prod_{i=1}^d g_i \cdot c_g} \right] \\
 &= E_{f(\mathbf{x})} \left[\log \prod_{i=1}^d \frac{f_i}{g_i} + \log \frac{c_f}{c_g} \right] \\
 &= \sum_{i=1}^d E_{f(\mathbf{x})} \left[\log \frac{f_i}{g_i} \right] + E_{f(\mathbf{x})} \left[\log \frac{c_f}{c_g} \right] \\
 &= \sum_{i=1}^d D_{KL}(f_i||g_i) + E_{f(\mathbf{x})} \left[\log \frac{c_f}{c_g} \right] .
 \end{aligned}$$

It is known that $u_i = F_i(x_i)$ for $i = 1, \dots, d$. By using a change of variables we have

$$\begin{aligned}
 E_{f(\mathbf{x})} \left[\log \frac{c_f}{c_g} \right] &= \int f(\mathbf{x}) \log \left(\frac{c_f}{c_g} \right) d\mathbf{x} \\
 &= \int \prod_{i=1}^d f_i \cdot c_f \log \left(\frac{c_f}{c_g} \right) \frac{1}{\prod_{i=1}^d f_i} d\mathbf{u} \\
 &= \int c_f \log \left(\frac{c_f}{c_g} \right) d\mathbf{u} \\
 &= D_{KL}(c_f||c_g) .
 \end{aligned}$$

Therefore,

$$D_{KL}(f||g) = \sum_{i=1}^d D_{KL}(f_i||g_i) + D_{KL}(c_f||c_g) .$$

□

Proposition 5.1 gives an encouragement for using copula functions. When a probabilistic model is proposed for a multivariate data set, the Kullback-Leibler divergence between the unknown density of the data set and the proposed density model depends on the selection of marginal densities and the copula function. Under the assumption that marginal densities are well selected, the proposed density differs only from the unknown density in the term related to the dependence among variables.

Proposition 5.2. *Let f be a d -dimensional density function with marginal densities f_i for $i = 1, \dots, d$. Then, the Kullback-Leibler divergence between the*

multivariate density f and the product of marginal densities $\prod_{i=1}^d f_i$ is given by the expression,

$$D_{KL} \left(f \parallel \prod_{i=1}^d f_i \right) = -H(U_1, \dots, U_d) \ ,$$

where $H(U_1, \dots, U_d)$ is the entropy of the copula function for the multivariate density f .

Proof. Using Proposition 5.1

$$\begin{aligned} D_{KL} \left(f \parallel \prod_{i=1}^d f_i \right) &= \sum_{i=1}^d D_{KL}(f_i \parallel f_i) + D_{KL}(c_f \parallel 1) \\ &= D_{KL}(c_f \parallel 1) \ . \end{aligned}$$

But,

$$\begin{aligned} D_{KL}(c_f \parallel 1) &= \int c_f \log \left(\frac{c_f}{1} \right) d\mathbf{u} \\ &= -H(U_1, \dots, U_d) \ . \end{aligned}$$

Thus,

$$D_{KL} \left(f \parallel \prod_{i=1}^d f_i \right) = -H(U_1, \dots, U_d) \ .$$

□

For the particular case of a two-dimensional random vector (X_1, X_2) , it is known that the Kullback-Leibler divergence between the bivariate density f and the product of marginal densities $f_1 \cdot f_2$ is equal to the mutual information between variables X_1 and X_2 . In this sense, Proposition 5.2 gives a theoretical support for the connection between mutual information and the entropy of a bivariate copula function presented in Davy and Doucet (2003) (see Equation (4.14)).

The following result was presented in Jenison and Reale (2004) and shows the relationship among the entropies of marginal densities, the entropy of the original distribution, and the entropy of the copula function.

Proposition 5.3. *Let f be a d -dimensional density function with marginal densities f_i for $i = 1, \dots, d$. Then, the entropy of the associated copula function is given by,*

$$H(U_1, \dots, U_d) = H(X_1, \dots, X_d) - \sum_{i=1}^d H(X_i) \ .$$

Proof. We first calculate the Kullback-Leibler divergence

$$\begin{aligned}
 D_{KL} \left(f \parallel \prod_{i=1}^d f_i \right) &= E_{f(\mathbf{x})} \left[\log \frac{f(\mathbf{x})}{\prod_{i=1}^d f_i} \right] \\
 &= E_{f(\mathbf{x})} [\log f(\mathbf{x})] - E_{f(\mathbf{x})} \left[\log \prod_{i=1}^d f_i \right] \\
 &= -H(X_1, \dots, X_d) + \sum_{i=1}^d H(X_i) .
 \end{aligned}$$

By using the result of Proposition 5.2, we complete the proof. \square

From the information theory, it is known that the sum of marginal entropies is greater or equal than the joint entropy. As a consequence, Proposition 5.3 states that the entropy of a copula function is non positive.

We present in the next proposition, two results for bivariate and trivariate dependence structures.

Proposition 5.4. *Let X_1, X_2 and X_3 be continuous random variables with joint and marginal densities. The following expressions hold for the mutual information and the conditional mutual information,*

$$\begin{aligned}
 I(X_1, X_2) &= I(U_1, U_2) , \\
 I(X_1, X_2|X_3) &= I(U_1, U_2|U_3) ,
 \end{aligned}$$

where U_1, U_2 and U_3 are the variables of the corresponding copula function.

Proof. By using Proposition 5.2 and the fact that marginal densities are uniform for copula functions,

$$\begin{aligned}
 I(X_1, X_2) &= D_{KL}(f \parallel f_1 \cdot f_2) \\
 &= D_{KL}(c_f \parallel 1) \\
 &= \int c_f \log(c_f) d\mathbf{u} \\
 &= I(U_1, U_2) .
 \end{aligned}$$

Similarly,

$$\begin{aligned}
 I(X_1, X_2|X_3) &= D_{KL}(f \parallel f_{1|3} \cdot f_{2|3} \cdot f_3) \\
 &= D_{KL}(c_f \parallel c(u_1, u_3) \cdot c(u_2, u_3)) \\
 &= \int c_f \log \left(\frac{c_f}{c(u_1, u_3) \cdot c(u_2, u_3)} \right) d\mathbf{u} \\
 &= I(U_1, U_2|U_3) .
 \end{aligned}$$

□

For calculating the mutual information, we can employ the bivariate copula entropy and Equation (4.14). For the case of the conditional mutual information, we can employ the relationship

$$I(X_1, X_2|X_3) = H(X_1) + H(X_2) + H(X_3) - H(X_1, X_2, X_3) - I(X_1, X_3) - I(X_2, X_3) , \quad (5.7)$$

and by doing the corresponding substitutions,

$$I(X_1, X_2|X_3) = -H(U_1, U_2, U_3) + H(U_1, U_3) + H(U_2, U_3) . \quad (5.8)$$

The results in Proposition 5.4 show that, under the assumption that marginal distributions are well fitted, the source of information about the dependence between variables can be calculated by using only copula functions. These results along with Equations (4.14) and (5.8) will be used for constructing the graphical structure of the D-vine EDA.

We illustrate how permutations of variables can modify the amount of information for each tree in a C-vine and a D-vine.

Example 5.2. Consider a four-dimensional Gaussian copula with correlation matrix given by

$$\Sigma = \begin{bmatrix} 1.00 & 0.61 & 0.62 & 0.39 \\ 0.61 & 1.00 & 0.47 & 0.50 \\ 0.62 & 0.47 & 1.00 & 0.49 \\ 0.39 & 0.50 & 0.49 & 1.00 \end{bmatrix}$$

Over all possible permutations of variables (U_1, U_2, U_3, U_4) , we build C-vines and D-vines. For each regular vine, we calculate the amount of information given for each tree. The amount of information is related to the bivariate copula functions in each tree. In this case, we calculate the mutual information associated to each bivariate copula. Tables 5.2 and 5.2 show the different contributions of a C-vine and a D-vine for the Gaussian copula.

It can be seen that by truncating the C-vine or the D-vine some piece of information can be lost. A motivation for truncating the C-vine or the D-vine is to reduce the complexity of the model and reduce the number of conditional bivariate copulas. However, it is a convenient procedure for choosing an adequate permutation without a huge loss of information. ◀

5.3 The proposed EDA

Description of the truncated D-vine

In order to show how a probabilistic model based on a D-vine can be used in EDAs we propose an approximation based on two trees, T_1 and T_2 . We define

Table 5.1: Amount of information given for each tree in a C-vine.

Permutation of roots	T_1	T_2	T_3
1 - 2 - 3 - 4	0.557762	0.080097	0.052155
1 - 2 - 4 - 3	0.557762	0.080097	0.052155
1 - 3 - 2 - 4	0.557762	0.073815	0.058437
1 - 3 - 4 - 2	0.557762	0.073815	0.058437
1 - 4 - 2 - 3	0.557762	0.131869	0.000383
1 - 4 - 3 - 2	0.557762	0.131869	0.000383
2 - 1 - 3 - 4	0.501336	0.136523	0.052155
2 - 1 - 4 - 3	0.501336	0.136523	0.052155
2 - 3 - 1 - 4	0.501336	0.18778	0.000898
2 - 3 - 4 - 1	0.501336	0.18778	0.000898
2 - 4 - 1 - 3	0.501336	0.066718	0.12196
2 - 4 - 3 - 1	0.501336	0.066718	0.12196
3 - 1 - 2 - 4	0.504671	0.126907	0.058437
3 - 1 - 4 - 2	0.504671	0.126907	0.058437
3 - 2 - 1 - 4	0.504671	0.184445	0.000898
3 - 2 - 4 - 1	0.504671	0.184445	0.000898
3 - 4 - 1 - 2	0.504671	0.07355	0.111793
3 - 4 - 2 - 1	0.504671	0.07355	0.111793
4 - 1 - 2 - 3	0.363622	0.32601	0.000383
4 - 1 - 3 - 2	0.363622	0.32601	0.000383
4 - 2 - 1 - 3	0.363622	0.204433	0.12196
4 - 2 - 3 - 1	0.363622	0.204433	0.12196
4 - 3 - 1 - 2	0.363622	0.2146	0.111793
4 - 3 - 2 - 1	0.363622	0.2146	0.111793

a class of density functions based on a truncated D-vine:

$$f_{\text{D-vine}}(\mathbf{x}) = \prod_{k=1}^d f(x_k) \prod_{i=1}^2 \prod_{j=1}^{d-i} c_{\gamma_j, \gamma_{j+i} | \gamma_{j+1}} , \quad (5.9)$$

where $\gamma = (\gamma_1, \dots, \gamma_d)$ is a permutation of the integers between 1 and d . Therefore, the d -dimensional density $f_{\text{D-vine}}(\mathbf{x})$ defined in Equation (5.9) is composed by the product of marginal densities and a copula density given by a D-vine with only two trees. Then, the goal is to choose a permutation $\gamma = (\gamma_1, \dots, \gamma_d)$ that minimizes the Kullback-Leibler divergence between the true density function $f(\mathbf{x})$ and the proposed density function $f_{\text{D-vine}}(\mathbf{x})$:

Table 5.2: Amount of information given for each tree in a D-vine.

Permutation tree T_1	T_1	T_2	T_3
1 - 2 - 3 - 4	0.494779	0.194337	0.000898
1 - 2 - 4 - 3	0.513812	0.054242	0.12196
1 - 3 - 2 - 4	0.511228	0.177889	0.000898
1 - 3 - 4 - 2	0.523704	0.054517	0.111793
1 - 4 - 2 - 3	0.351145	0.216909	0.12196
1 - 4 - 3 - 2	0.344588	0.233633	0.111793
2 - 1 - 3 - 4	0.61255	0.019027	0.058437
2 - 1 - 4 - 3	0.452468	0.237164	0.000383
2 - 3 - 1 - 4	0.449883	0.181695	0.058437
2 - 3 - 4 - 1	0.344588	0.233633	0.111793
2 - 4 - 1 - 3	0.468916	0.220715	0.000383
2 - 4 - 3 - 1	0.523704	0.054517	0.111793
3 - 1 - 2 - 4	0.619107	0.018752	0.052155
3 - 1 - 4 - 2	0.468916	0.220715	0.000383
3 - 2 - 1 - 4	0.439991	0.197868	0.052155
3 - 2 - 4 - 1	0.351145	0.216909	0.12196
3 - 4 - 1 - 2	0.452468	0.237164	0.000383
3 - 4 - 2 - 1	0.513812	0.054242	0.12196
4 - 1 - 2 - 3	0.439991	0.197868	0.052155
4 - 1 - 3 - 2	0.449883	0.181695	0.058437
4 - 2 - 1 - 3	0.619107	0.018752	0.052155
4 - 2 - 3 - 1	0.511228	0.177889	0.000898
4 - 3 - 1 - 2	0.61255	0.019027	0.058437
4 - 3 - 2 - 1	0.494779	0.194337	0.000898

$$\begin{aligned}
 D_{KL}(f||f_{\text{D-vine}}) &= E_{f(\mathbf{x})} \left[\log \frac{f(\mathbf{x})}{f_{\text{D-vine}}(\mathbf{x})} \right] \\
 &= -H(\mathbf{X}) + \sum_{k=1}^d H(X_k) - \sum_{i=2}^d I(X_{\gamma_{(i-1)}}, X_{\gamma_i}) \\
 &\quad - \sum_{i=3}^d I(X_{\gamma_{(i-2)}}, X_{\gamma_i} | X_{\gamma_{(i-1)}}) , \tag{5.10}
 \end{aligned}$$

where $H(X_k)$ denotes the entropy of the continuous random variable X_k , $I(X_{\gamma_{(i-1)}}, X_{\gamma_i})$ is the marginal mutual information between $(X_{\gamma_{(i-1)}}, X_{\gamma_i})$ and $I(X_{\gamma_{(i-2)}}, X_{\gamma_i} | X_{\gamma_{(i-1)}})$ is the conditional mutual information between $(X_{\gamma_{(i-2)}}, X_{\gamma_i})$ given $X_{\gamma_{(i-1)}}$. The first two terms in the divergence do not depend on γ . There-

fore, minimizing the Kullback-Leibler is equivalent to maximizing

$$J_{\text{D-vine}}(\mathbf{X}) = \sum_{i=2}^d I(X_{\gamma_{(i-1)}}, X_{\gamma_i}) + \sum_{i=3}^d I(X_{\gamma_{(i-2)}}, X_{\gamma_i} | X_{\gamma_{(i-1)}}) . \quad (5.11)$$

It can be proved that there is a close relationship between the mutual information and the copula entropy. Equation (4.14) can be used in order to calculate marginal mutual information of any two arbitrary variables S and T .

An important consequence for bivariate Gaussian copulas is that, by definition, its entropy is equal to the negative of mutual information of two variables with standard joint Gaussian distribution

$$H(U_1, U_2) = \frac{1}{2} \log(1 - \theta^2) , \quad (5.12)$$

where θ is the correlation parameter.

For three arbitrary variables (X_1, X_2, X_3) , we state that

$$\begin{aligned} H(U_1, U_2, U_3) &= -I(X_1, X_3 | X_2) - I(X_1, X_2) - I(X_2, X_3) \\ &= -I(X_1, X_2, X_3) \\ &= -D_{KL}(f(x_1, x_2, x_3) || f_1 f_2 f_3) . \end{aligned} \quad (5.13)$$

The result in Equation (5.13) implies that the entropy of a trivariate Gaussian copula is given by:

$$H(U_1, U_2, U_3) = \frac{1}{2} \log(1 + 2\theta_{12}\theta_{13}\theta_{23} - \theta_{12}^2 - \theta_{13}^2 - \theta_{23}^2) . \quad (5.14)$$

The optimal permutation γ is the one that produces the highest marginal and conditional pairwise mutual information with respect to the true distribution. But due to computational efficiency reasons we will employ a greedy algorithm based on (De Bonet et al., 1997). We select the three variables with the smallest copula entropy (5.14) and choose a random order to make a chain. The following variables of γ are chosen according to their mutual information with respect to any of the variables in the ends of the chain. Algorithm 10 shows a straightforward greedy algorithm to find a permutation γ .

Algorithm 10 Greedy algorithm to pick a permutation γ in a D-vine

- 1: Find $(\gamma_{m-1}, \gamma_m, \gamma_{m+1}) = \arg \min_{j \neq k \neq l} \widehat{H}(U_j, U_k, U_l)$, where $\widehat{H}()$ is an estimation of the Gaussian copula entropy among variables $u_j = F_{X_j}(x_j)$, $u_k = F_{X_k}(x_k)$, and $u_l = F_{X_l}(x_l)$.
 - 2: Choose variables with the greatest mutual information with respect to any of the ends of the chain. The constraint is to avoid a circular chain.
 - 3: The order of the chain defines permutation γ .
-

5.3.1 Incorporating a new graphical model

In (Salinas-Gutiérrez et al., 2010c), a regular vine (Bedford and Cooke, 2001, 2002) is considered as a graphical model for designing a new EDA. This EDA is based on a particular family of regular vines called *D-vine*. In this work we propose a greedy algorithm for building such graphical model and we also show the theoretical relationship between the entropy of a trivariate copula function and the conditional mutual information.

Description of the D-vine ^{Gaussian}_{Beta} EDA

The probabilistic model used by the D-vine EDA has been previously presented in subsection 5.3. Once a permutation γ is found, generating samples follows the steps of Algorithm 7. We summarize the proposed approach in Algorithm 11.

Algorithm 11 Pseudocode for estimating the model and generating a new population

```

1: for  $j = 1 \rightarrow d$  do
2:   For each variable  $X_j$ , estimate its marginal Beta parameters  $(a_j, b_j)$ .
3:   Determine  $U_j = F_j(X_j; a_j, b_j)$ , where  $F_j$  is the cumulative Beta distribution function.
4: end for
5: Estimate the parameters of the Gaussian copula  $\Sigma$  using Algorithm 3.
6: Calculate all bivariate and trivariate copula entropies, Equations (5.12) and (5.14).
7: Pick a permutation  $\gamma$  for the graphical model using Algorithm 10.
8: Simulate  $U_{\gamma_1}$  from a uniform distribution  $U(0, 1)$ .
9: Simulate  $U_{\gamma_2}$  from the conditional Gaussian copula  $C(U_{\gamma_2}|U_{\gamma_1})$ .
10: for  $k = 3 \rightarrow d$  do
11:   Simulate  $U_{\gamma_k}$  from the conditional Gaussian copula  $C(U_{\gamma_k}|U_{\gamma_{k-2}}, U_{\gamma_{k-1}})$ .
12: end for
13: for  $j = 1 \rightarrow d$  do
14:   Determine  $X_j$  using quasi-inverse  $F_j^{-1}(U_j)$ .
15: end for

```

It is important to say that the graphical model used in the algorithm MIMIC is a particular case of the D-vine EDA when there is no tree T_2 . The graphical model used in the algorithm UMDA is also a particular case of the D-vine EDA when there are no trees T_1 and T_2 .

For each decision variable in the proposed algorithm, the Beta distribution is used as marginal distribution. The test problems used in this subsection have bounded search space. Each value of variable X_i from the search space is transformed to a value in $(0, 1)$ through a linear transformation. This explains why we use Beta distributions as marginals.

Given that the proposed algorithm uses a D-vine graphical model, Gaussian

copulas, and Beta marginal distributions, a specific notation employed in this thesis for the D-vine EDA is D-vine $\text{Gaussian}_{\text{Beta}}$.

Experiments

Algorithms MIMIC_c^G , UMDA_c^G along with the proposed algorithm D-vine $\text{Gaussian}_{\text{Beta}}$, are used to optimize three test problems. The probabilistic model in MIMIC_c^G is built using Algorithm 10 and changing conditional mutual information to marginal mutual information in Step 1. The test functions Ackley, Rosenbrock, and Sphere are used in the experiments. The test functions are described in Appendix B. We use test problems in 10 dimensions. Each algorithm is run 30 times for each problem. The population size is 300 and the number of selected individuals is 200. The maximum number of evaluations is 300,000. However, when convergence to a local minimum is detected, the run is stopped. Any improvement less than 1×10^{-6} in 25 iterations is considered convergence. The goal is to reach the optimum with an error less than 1×10^{-6} .

Numerical results

In Table 5.3 we report the fitness value reached by the algorithms in all test functions. The information about the number of evaluations required by each algorithm is reported in Table 5.4.

Table 5.3: Descriptive fitness results for all test functions.

Algorithm	Best	Median	Mean	Worst	Std. deviation
Ackley					
D-vine $\text{Gaussian}_{\text{Beta}}$	2.71E-005	4.11E-005	4.09E-005	5.16E-005	5.82E-006
MIMIC_c^G	2.89E-005	3.91E-005	3.92E-005	4.95E-005	5.21E-006
UMDA_c^G	2.83E-005	4.08E-005	4.15E-005	5.07E-005	4.63E-006
Rosenbrock					
D-vine $\text{Gaussian}_{\text{Beta}}$	0.20	5.75	5.28	9.21	2.70
MIMIC_c^G	0.91	6.40	6.42	13.48	2.72
UMDA_c^G	7.93	8.02	8.10	10.28	0.42
Sphere					
D-vine $\text{Gaussian}_{\text{Beta}}$	3.37E-007	7.75E-007	7.48E-007	9.99E-007	1.69E-007
MIMIC_c^G	4.48E-007	8.47E-007	8.14E-007	9.96E-007	1.52E-007
UMDA_c^G	4.20E-007	8.56E-007	8.11E-007	9.94E-007	1.64E-007

To properly compare the performance of the algorithms (using the optimum value reached), we conduct a hypothesis test based on a Bootstrap method for the differences between the means of the three comparison pairs, for all test problems. Table 5.5 shows the confidence intervals for the means, and the corresponding p-values.

Table 5.4: Descriptive function evaluations for all test functions.

Algorithm	Mean	Std. deviation
Ackley		
D-vine ^{Gaussian} _{Beta}	74691.20	3810.83
MIMIC _c ^G	77063.27	4773.64
UMDA _c ^G	76873.90	3924.77
Rosenbrock		
D-vine ^{Gaussian} _{Beta}	239797.47	111039.69
MIMIC _c ^G	223295.83	119242.76
UMDA _c ^G	291245.87	47948.36
Sphere		
D-vine ^{Gaussian} _{Beta}	65402.27	905.32
MIMIC _c ^G	65920.53	960.83
UMDA _c ^G	67624.83	1099.53

Discussion

According to Table 5.3, the Ackley problem is solved in a similar way by the three algorithms. Table 5.5 shows no differences in algorithms at a significance level $\alpha = 0.05$. However, if we consider a significance level of $\alpha = 0.08$, the p-value shows a significant difference between MIMIC_c^G and UMDA_c^G. This would mean that a dependence structure based on a MIMIC_c^G is more adequate than an independence structure.

For the Rosenbrock problem, the algorithm that shows the best behaviour is D-vine EDA. Although D-vine EDA has a standard deviation greater than UMDA_c^G, we can see in Table 5.3 that the range of values reached by D-vine is closer to the global minimum than the range of values of UMDA_c^G. P-values indicate that D-vine EDA and MIMIC_c^G are statistically different from UMDA_c^G. In these cases, the confidence intervals between D-vine EDA vs. UMDA_c^G and MIMIC_c^G vs. UMDA_c^G show that better results are obtained using dependence structure. However, if we consider a significance level of $\alpha = 0.11$, the p-value shows a significant difference between D-vine EDA and MIMIC_c^G.

D-vine, MIMIC_c^G and UMDA_c^G obtain the global minimum in all the runs for the Sphere Model problem. There are no statistical differences between all the algorithms and also the number of evaluations is similar. This means that the univariate EDA is the adequate tool for the Sphere problem.

Conclusions

In this subsection, we have introduced the regular vine graphical models in EDAs. According to numerical experiments the selection of a graphical model for modeling dependence structure can help achieve better fitness results. D-vine EDA obtain a better performance than MIMIC and UMDA in a well known hard optimization problem such as the Rosenbrock function. This means that

Table 5.5: Results for the difference between fitness means in each problem. A 95% confidence interval and a p-value are obtained through a Bootstrap technique.

Compared algorithms	95% Interval	p-value
Ackley		
D-vine ^{Gaussian} _{Beta} vs. MIMIC _c ^G	-1.0E-06	4.4E-06
D-vine ^{Gaussian} _{Beta} vs. UMDA _c ^G	-3.2E-06	2.0E-06
MIMIC _c ^G vs. UMDA _c ^G	-4.7E-06	2.0E-07
Rosenbrock		
D-vine ^{Gaussian} _{Beta} vs. MIMIC _c ^G	-2.5	1.7E-01
D-vine ^{Gaussian} _{Beta} vs. UMDA _c ^G	-3.8	-1.9
MIMIC _c ^G vs. UMDA _c ^G	-2.6	-6.9E-01
Sphere		
D-vine ^{Gaussian} _{Beta} vs. MIMIC _c ^G	-1.4E-07	1.5E-08
D-vine ^{Gaussian} _{Beta} vs. UMDA _c ^G	-1.4E-07	2.0E-08
MIMIC _c ^G vs. UMDA _c ^G	-7.6E-08	8.1E-08

dependencies between decision variables must be modeled adequately in order to get good solutions.

UMDA does not outperform D-vine EDA and MIMIC in any optimization problem. The UMDA results are similar to D-vine EDA and MIMIC in the Sphere Model. This is because Sphere Model is not a hard optimization problem and it is more adequate for EDAs without dependence models. All three algorithms perform in a similar way when solving the Ackley function. In this case one may be tempted to quickly say that a complex probabilistic model was outperformed by a more simple model such as the UMDA. However, a smarter learning algorithm for the D-vine graphical model could build such simpler models.

Although we use the same copula function and the same marginal distribution for the proposed EDA, it is not necessary. We state that fitness results in a EDA are the consequence of the selected marginal distributions, copula functions and graphical model.

5.4 Summary

In this chapter, we have presented the incorporation of a new graphical model into continuous EDAs. Theoretical results concerning measures such as entropy and mutual information along with copula functions have been provided for designing a truncated D-vine.

Chapter 6

Conclusions

This dissertation uses elements and methods from information theory, graphical models, and copula theory for designing multivariate distributions and applying them into optimization problems. Our approach has been to model the most important dependencies in the selected population and to estimate their associated parameters in the corresponding multivariate distribution.

Chapter 3 provides a computational introduction to the copula theory, while chapter 4 describes how copula functions can be integrated into graphical models for learning a multivariate distribution of the dependence structure among decision variables of the selected population. The approach followed in this thesis has been mainly motivated by the possibility of proposing novel probabilistic models in EDAs.

This doctoral work has been conducted for investigating the incorporation of copula functions as probabilistic models into continuous EDAs. Although there have been published other works related to the application of copula theory in continuous EDAs, this doctoral dissertation presents the following contributions:

- The estimation of copula parameters. Some related works that incorporate copula functions in EDAs use predefined values for the parameters, however, it makes no sense in most practical cases. Our contribution for estimating the parameters of the copula functions has been to use the maximum likelihood method. This method is a well established procedure for doing statistical inference with a solid theoretical support. This model fitting for the copula functions has been done since the initial stage of our research work.
- The use of copula entropies for building the graphical model. For most of the EDAs, the learning of a predefined structure in the probabilistic model is made by minimizing the Kullback-Leibler divergence. In particular, the learning of structures such as a chain and a tree is done by the minimization of the total sum of the conditional entropies. This way of learning the structure of the probabilistic model is affected by the dependence among variables and the behavior of marginal distributions. Our

proposed algorithms can learn the structure of the probabilistic model only by taking into account the dependence among variables, regardless the marginal distributions. It can be done by using the copula entropy as a measure of dependence. Moreover, in this research, we have provided the relationship between the entropy of a copula function and the measure known as mutual information.

- Flexibility for choosing the marginal distributions. Nowadays, the research on continuous EDAs has been mainly based on the use of the multivariate normal distribution as probabilistic model. A direct consequence is that each marginal distribution is restricted to be of the same type, i.e., be an univariate normal distribution. We have described in this thesis how the separation between marginal distributions and a dependence structure explains the modeling flexibility given by copula functions.
- Flexibility for modeling the dependence among variables. Given that most of the literature for continuous EDAs has explored the use of the multivariate normal distributions as probabilistic models, only linear dependencies can be modeled. The assumption of modeling dependencies under the multivariate normal distribution can not be realistic for some optimization problems. By means of copula functions, a wide range of dependencies can be incorporated to our proposed algorithms.
- An extension for some continuous EDAs. Several EDAs that employ the multivariate normal distribution assumption can be seen as particular instances of our proposed algorithms. They can be obtained by modeling the dependence structure with a Gaussian copula and by using, for each decision variable, the univariate normal distribution as marginal distribution.
- A first study about the natural connection between copula functions and graphical models. Besides the previous contributions, we have investigated the promising idea of the incorporation of copula functions into graphical models. A practical consequence is that any joint distribution can be factorized by copula functions of lower order, with no restriction over the class of copula functions and the family of marginal distributions. Other consequences are that the most important dependencies are represented by the graphical model and these dependencies can be linear or nonlinear.
- One recent contribution of our research work has been the incorporation of a procedure for selecting the most adequate copula function.
- A step towards the learning of the problem structure. An important goal of modeling dependencies among variables in an EDA is to learn the structure of the optimization problem. The learning of the problem structure by means of a probabilistic model can help ensure an efficient optimization behavior.

-
- A new research direction in EDAs. This work has opened a new research direction for continuous EDAs. The incorporation of copula functions in continuous EDAs represents a promising opportunity for designing more flexible EDAs and for investigating their performance.

This thesis has concentrated on investigating the incorporation of copula functions into continuous EDAs. The performance of the proposed EDAs has been investigated empirically and the theoretical support of the probabilistic models have been presented. However, it is necessary to provide some critical observations regarding them. The idea of using a probabilistic model in EDAs is that the structural features of an optimization problem can be captured and exploited, in order to efficiently find an optimum value. This shows the following drawbacks:

- The structure of a given optimization problem is unknown. In general, it is not possible to select the most suitable probabilistic model before applying it in an EDA. Thus, the performance of an EDA depends on how well the proposed probabilist model emulates the structure of the optimization problem.
- Copula selection. Although we have provided in this thesis the incorporation of a procedure for selecting copula functions, this does not ensure that the overfitting is avoided. It is well known, (Bosman, 2003), that selecting a model based on the maximum likelihood is equivalent to select it with minimal entropy. Therefore, it is convenient to prevent the possibility of losing the ability to generalize the data through a copula function.

A general limitation of using our proposed algorithms is given by the *No Free Lunch theorem* (NFL). Informally speaking, this theorem states that the average performance for all optimization algorithms over all possible problems is the same. So, there is no the best optimizer for *all* problems. However, for a given optimization problem, the NFL can be circumvent by incorporating knowledge about the problem.

Conclusions regarding the performance of the proposed EDAs have already been discussed in chapter 4. The most important ones are briefly commented next. According to our experiments, the performance of the copula based EDA strongly depends on the selected marginal distributions, copula functions and the graphical model. This shows that dependencies between decision variables must be modeled adequately in order to get good solutions. The EDA based on a chain graphical model and the EDA based on a tree graphical model, both with a copula selection procedure, have a better performance than EDAs based on multivariate Gaussian distributions. The success rate already indicates a better performance of the algorithms adapted with copula selection in higher dimension. The obtained performance results suggest that the incorporation of copula functions is a real option for designing new continuous EDAs. Moreover, our reported experiments have shown that modeling adequately the dependencies between decision variables is a good mechanism for getting better performance

results. Furthermore, the presented methods for learning probabilistic models can be applied to other problems not necessarily related to optimization. For example, in (Salinas-Gutiérrez et al., 2010a,b) we have applied Gaussian copulas to classification problems.

Our research is a first step towards the almost unexplored field of EDAs based on copula functions. We believe that there are many directions for future research. For example, it would be interesting to further explore the connections between graphical models, copula functions and concepts from information theory. More experiments are necessary with different probabilistic models in order to identify the class of function properties where the copula functions provide advantages to EDAs. Given that all the algorithms presented in this dissertation estimate their probabilistic parameters only by means of maximum likelihood, an immediate research work will focus on the design of algorithms with diversity maintenance, or similar strategies, for enhancing the performance. A study of the adaptation and performance of EDAs based on copula functions is also necessary for solving problems from multiobjective optimization and constrained optimization. Finally, the copula functions that we have used for designing probabilistic models are a small part of the known set of copula functions. In particular, more Archimedean copulas can be investigated and nonparametric copula functions can be also considered.

Bibliography

- K. Aas, C. Czado, A. Frigessi, and H. Bekken. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44(2):182–198, April 2009. 58
- R.J. Arderí-García. *Algoritmo con Estimación de Distribuciones con Cópula Gaussiana*. Universidad de La Habana, La Habana, Cuba, June 2007. Bachelor’s thesis, in Spanish. 5
- T. Bacigál and M. Komorníková. Fitting Archimedean copulas to bivariate geodetic data. In A. Rizzi and M. Vichi, editors, *Compstat 2006 Proceedings in Computational Statistics*, pages 649–656, Heidelberg, Germany, 2006. Physica-Verlag HD. ISBN 978-3-7908-1708-9. doi: 10.1007/978-3-7908-1709-6. 2, 17
- S. Baluja. Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, PA, USA, June 1994. 4, 9
- S. Baluja and R. Caruana. Removing the Genetics from the Standard Genetic Algorithm. In A. Prieditis and S. Russell, editors, *Proceedings of the Twelfth International Conference on Machine Learning*, pages 38–46. Morgan Kaufmann, 1995. 9
- S. Baluja and S. Davies. Using Optimal Dependency-Trees for Combinatorial Optimization: Learning the Structure of the Search Space. In D.H. Fisher, editor, *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 30–38. Morgan Kaufmann, 1997. 4, 11, 33
- S.E. Barba-Moreno. Una propuesta para EDAs no paramétricos. Master’s thesis, Centro de Investigación en Matemáticas, Guanajuato, México, December 2007. In Spanish. 4
- T. Bedford and R. M. Cooke. Probability Density Decomposition for Conditionally Dependent Random Variables Modeled by Vines. *Annals of Mathematics and Artificial Intelligence*, 32(1):245–268, January 2001. ISSN 1012-2443. doi: 10.1023/A:1016725902970. 57, 67

- T. Bedford and R. M. Cooke. Vines – A New Graphical Model for Dependent Random Variables. *The Annals of Statistics*, 30(4):1031–1068, August 2002. 57, 67
- E. Bengoetxea. *Inexact Graph Matching Using Estimation of Distribution Algorithms*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, France, December 2002. 4
- P.A.N. Bosman. *Design and Application of Iterated Density-Estimation Evolutionary Algorithms*. PhD thesis, University of Utrecht, Utrecht, The Netherlands, 2003. 4, 33, 73
- P.A.N. Bosman and J. Grahl. Matching Inductive Search Bias and Problem Structure in Continuous Estimation of Distribution Algorithms. Technical Report 03/2005, University of Mannheim, Mannheim, Germany, 2005. 2
- P.A.N. Bosman, J. Grahl, and D. Thierens. Enhancing the Performance of Maximum-Likelihood Gaussian EDAs Using Anticipated Mean Shift. In G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume, editors, *Parallel Problem Solving from Nature – PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 133–143. Springer Berlin / Heidelberg, 2008. 4
- N. Brunel, W. Pieczynski, and S. Derrode. Copulas in vectorial hidden markov chains for multicomponent image segmentation. In *ICASSP '05: Proceedings of the 2005 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 717–720, 2005. 17
- U. Cherubini, E. Luciano, and W. Vecchiato. *Copula Methods in Finance*. Wiley, Chichester, 2004. 2, 17, 21, 27, 42
- C.K. Chow and C.N. Liu. Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Transactions on Information Theory*, 14(3): 462–467, May 1968. 11, 37
- T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. Wiley, first edition, 1991. 36
- A. Cuesta-Infante, R. Santana, J.I. Hidalgo, C. Bielza, and P. Larrañaga. Bivariate empirical and n-variate Archimedean copulas in Estimation of Distribution Algorithms. In *WCCI 2010 IEEE World Congress on Computational Intelligence*, pages 1355–1362, July 2010. 5
- M. Davy and A. Doucet. Copulas: a new insight into positive time-frequency distributions. *Signal Processing Letters, IEEE*, 10(7):215–218, 2003. doi: 10.1109/LSP.2003.811636. 38, 41, 61
- J.S. De Bonet, C.L. Isbell, and P. Viola. MIMIC: Finding Optima by Estimating Probability Densities. In *Advances in Neural Information Processing Systems*, volume 9, pages 424–430. The MIT Press, 1997. 4, 10, 33, 66

-
- D.J. De-Waal and P.H.A.J.M. Van-Gelder. Modelling of extreme wave heights and periods through copulas. *Extremes*, 8(4):345–356, 2005. ISSN 1386-1999. doi: 10.1007/s10687-006-0006-y. 2, 17
- K. Dowd. Copulas in Macroeconomics. *Journal of International and Global Economic Studies*, 1(1):1–26, 2008. 17
- R. Etxeberria and P. Larrañaga. Global optimization with Bayesian networks. In A. Ochoa, M. Soto, and R. Santana, editors, *Second International Symposium on Artificial Intelligence. Adaptive Systems. CIMAF99*, pages 332–339, La Habana, 1999. Academia. ISBN 959-02-0241-1. 4, 13
- F. Flitti, C. Collet, and Joannic-Chardin A. Unsupervised Multiband Image Segmentation using Hidden Markov Quadtree and Copulas. In *IEEE International Conference on Image Processing*, Genova, Italy, Sep 2005. URL <http://www.icip05.org/>. 17
- E. Flores de la Fuente. EDAs con Funciones de cópula. Master’s thesis, Centro de Investigación en Matemáticas, Guanajuato, México, September 2009. In Spanish. 5
- E. W. Frees and E. A. Valdez. Understanding relationships using copulas. *North American Actuarial Journal*, 2(1):1–25, January 1998. 17
- Y. Gao. Multivariate Estimation of Distribution Algorithm with Laplace Transform Archimedean Copula. In W. Hu and X. Li, editors, *2009 International Conference on Information Engineering and Computer Science, ICIECS 2009*, Wuhan, China, December 2009. 5
- C. Genest and A.C. Favre. Everything You Always Wanted to Know about Copula Modeling but Were Afraid to Ask. *Journal of Hydrologic Engineering*, 12(4):347–368, July 2007. 2, 17
- C. González. *Contributions on Theoretical Aspects of Estimation of Distribution Algorithms*. PhD thesis, University of the Basque Country, Donostia-San Sebastián, Spain, November 2005. 3, 4
- J. Grahl, S. Minner, and P.A.N. Bosman. Learning structure illuminates black boxes – an introduction into Estimation of Distribution Algorithms. Technical Report 10/2006, University of Mannheim, Mannheim, Germany, 2006. 2
- M. Grigoriu. Multivariate distributions with specified marginals: Applications to Wind Engineering. *Journal of Engineering Mechanics*, 133(2):174–184, 2007. 17
- L. Grosset. *Optimization of Composite Structures by Estimation of Distribution Algorithms*. PhD thesis, University of Florida, Florida, United States, 2004. 4

- G. Harik. Linkage Learning via Probabilistic Modeling in the ECGA. Technical Report 99010, University of Illinois, Urbana, Illinois, 1999. 12
- G. Harik, F.G. Lobo, and D.E. Goldberg. The Compact Genetic Algorithm. In *Proceedings of the IEEE Conference on Evolutionary Computation*, pages 523–528, 1998. 4, 9
- M. Henrion. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In J.F. Lemmer and L.N. Kanal, editors, *Uncertainty in Artificial Intelligence*, volume 2, pages 149–163. Elsevier Science Ltd, 1988. 40
- R. Höns. *Estimation of Distribution Algorithm and Minimum Relative Entropy*. PhD thesis, University of Bonn, Bonn, Germany, 2005. 4
- C. Igel, T. Sutton, and N. Hansen. A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation, GECCO '06*, pages 453–460. ACM, 2006. 4
- K. Jajuga and D. Papla. Copula Functions in Model Based Clustering. In M. Spiliopoulou, R. Kruse, C. Borgelt, A. Nürnberger, and W. Gaul, editors, *From Data and Information Analysis to Knowledge Engineering*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 606–613. Springer Berlin Heidelberg, 2006. 17
- R.L. Jenison and R.A. Reale. The Shape of Neural Dependence. *Neural Computation*, 16:665–672, 2004. 61
- F.B. Jensen and T.D. Nielsen. *Bayesian Networks and Decision Graphs*. Information Science and Statistics. Springer, 2007. 41
- H. Joe. *Multivariate models and dependence concepts*. Chapman & Hall, London, 1997. 18
- D. Kurowicka and R. Cooke. *Uncertainty Analysis*. Wiley Series in Probability and Statistics. Wiley, 2006. 57, 58
- D. Kurowicka and R. Cooke. The vine copula method for representing high dimensional dependent distributions: application to continuous belief nets. In E. Yücesan, C.-H. Chen, J.L. Snowdon, and J.M. Charnes, editors, *Proceedings of the 2002 Winter Simulation Conference*, pages 270–278, 2002. 55
- P. Larrañaga, R. Etxeberria, J.A. Lozano, and J.M. Peña. Optimization by learning and simulation of Bayesian and Gaussian networks. Technical Report EHU-KZAA-IK-4/99, Department of Computer Science and Artificial Intelligence, University of the Basque Country, 1999. 4, 8, 10, 14, 33, 41
- P. Larrañaga, R. Etxeberria, J.A. Lozano, and J.M. Peña. Optimization in continuous domains by learning and simulation of Gaussian networks. In *Proceedings of the Optimization by Building and Using Probabilistic Models*

-
- OBUPM Workshop at the Genetic and Evolutionary Computation Conference GECCO-2000*, pages 201–204, 2000a. 4, 8, 10, 14, 33, 41
- P. Larrañaga, R. Etxeberria, J.A. Lozano, and J.M. Peña. Combinatorial optimization by learning and simulation of Bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 343–352, 2000b. 4, 13
- P. Larrañaga, J.A. Lozano, and E. Bengoetxea. Estimation of Distribution Algorithm based on multivariate normal and Gaussian networks. Technical Report EHU-KZAA-IK-1/01, Department of Computer Science and Artificial Intelligence, University of the Basque Country, 2001. 4, 14, 33
- P. Larrañaga. *An Introduction to Probabilistic Graphical Models*, chapter 2, pages 27–56. In , Larrañaga and Lozano (2002), 2002a. 41
- P. Larrañaga. *A Review on Estimation of Distribution Algorithms*, chapter 3, pages 57–100. In , Larrañaga and Lozano (2002), 2002b. 3
- P. Larrañaga and J.A. Lozano, editors. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Genetic Algorithms and Evolutionary Computation. Kluwer Academic Publishers, 2002. 4, 79
- S.L. Lauritzen. *Graphical Models*. Oxford, 1996. 34
- A. Mendiburu-Alberro. *Parallel implementation of Estimation of Distribution Algorithms based on probabilistic graphical models. Application to chemical calibration models*. PhD thesis, University of the Basque Country, Donostia-San Sebastián, Spain, January 2006. 4
- G. Mercier, L. Bouchemakh, and Y. Smara. The Use of Multidimensional Copulas to Describe Amplitude Distribution of Polarimetric SAR Data. In *IGARSS 07*, 2007. 17
- P.E. Monjardin. Análisis de dependencia en tiempo de falla. Master’s thesis, Centro de Investigación en Matemáticas, Guanajuato, México, December 2007. In Spanish. 2, 17
- H. Mühlenbein. The Equation for Response to Selection and its Use for Prediction. *Evolutionary Computation*, 5(3):303–346, 1998. 3, 4
- H. Mühlenbein and T. Mahnig. FDA - a scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(4):353–376, 1999. 13
- H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. In W. Ebeling, I. Rechenberg, H.M. Voigt, and H.P. Schwefel, editors, *Proceedings of the 4th Conference on Parallel Problem Solving from Nature*, number 1141 in Lecture Notes in Computer Science, pages 178–187. Springer-Verlag, 1996. 3, 8

- H. Mühlenbein, T. Mahnig, and A. Ochoa-Rodriguez. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5: 215–247, 1999. 3
- R.E. Neapolitan. *Learning Bayesian Networks*. Series in Artificial Intelligence. Prentice Hall, 2004. 41
- R.B. Nelsen. *An Introduction to Copulas*. Springer Series in Statistics. Springer-Verlag, second edition, 2006. 18, 20, 21, 24, 31
- J. Očenášek. *Parallel Estimation of Distribution Algorithms*. PhD thesis, Brno University of Technology, Czech Republic, November 2002. 4
- T.K. Paul and H. Iba. Reinforcement Learning Estimation of Distribution Algorithm. In E. Cantú-Paz, J. Foster, K. Deb, L. Davis, R. Roy, U.M. O’Reilly, H.G. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. Potter, A. Schultz, K. Dowsland, N. Jonoska, and J. Miller, editors, *Proceedings of the Genetic and Evolutionary Computation Conference 2003 (GECCO 2003)*, volume 2724 of *Lecture Notes in Computer Science*, pages 1259–1270. Springer Berlin / Heidelberg, 2003. 10
- M. Pelikan. *Bayesian optimization algorithm: From single level to hierarchy*. PhD thesis, University of Illinois at Urbana-Champaign, Illinois, United States, 2002. 4
- M. Pelikan and H. Mühlenbein. The Bivariate Marginal Distribution Algorithm. In R. Roy, T. Furuhashi, and P.K. Chawdhry, editors, *Advances in Soft Computing - Engineering Design and Manufacturing*, pages 521–535, London, 1999. Springer-Verlag. ISBN 1-85233-062-7. 4, 11, 33
- M. Pelikan, D.E. Goldberg, and E. Cantú-Paz. BOA: The Bayesian Optimization Algorithm. In W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, and R.E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, volume 1, pages 525–532. Morgan Kaufmann Publishers, 1999. 4, 13, 33
- M. Pelikan, K. Sastry, and E. Cantú-Paz, editors. *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, volume 33 of *Studies in Computational Intelligence*. Springer, 2006. 4
- D. Romero-Asturiano. *Algoritmos de Estimación de Distribuciones Aplicados a Problemas Combinatorios en Modelos Gráficos Probabilísticos*. PhD thesis, University of the Basque Country, Donostia-San Sebastián, Spain, March 2007. In Spanish. 4
- S. Rudlof and M. Köppen. Stochastic Hill Climbing with Learning by Vectors of Normal Distributions. In T. Furuhashi, editor, *Proceedings of the First Online Workshop on Soft Computing (WSC1)*, Nagoya, Japan, 1996. 9

-
- S. Sakji-Nsibi and A. Benazza-Benyahia. Multivariate indexing of multichannel images based on the copula theory. In *IPTA08*, 2008. 17
- R. Salinas Gutiérrez, A. Hernández Aguirre, and E.R. Villa Diharce. Una extensión del algoritmo MIMIC mediante Cópulas. *Revista Electrónica Nova Scientia*, 2(3):1–13, 2009. ISSN 2007-0705. URL http://novascientia.delasalle.edu.mx/numero_3/index.html. In Spanish. 5, 41
- R. Salinas-Gutiérrez, A. Hernández-Aguirre, and E.R. Villa-Diharce. Using Copulas in Estimation of Distribution Algorithms. In A. Hernández Aguirre, R. Monroy Borja, and C.A. Reyes García, editors, *MICAI 2009: Advances in Artificial Intelligence*, volume 5845 of *Lectures Notes in Artificial Intelligence*, pages 658–668. Springer, 2009. 5, 17, 41
- R. Salinas-Gutiérrez, A. Hernández-Aguirre, M.J.J. Rivera-Meraz, and E.R. Villa-Diharce. Using Gaussian Copulas in Supervised Probabilistic Classification. In O. Castillo, J. Kacprzyk, and W. Pedrycz, editors, *Soft Computing for Intelligent Control and Mobile Robotics*, volume 318 of *Studies in Computational Intelligence*, pages 355–372. Springer Berlin / Heidelberg, 2010a. 18, 74
- R. Salinas-Gutiérrez, A. Hernández-Aguirre, M.J.J. Rivera-Meraz, and E.R. Villa-Diharce. Supervised Probabilistic Classification Based on Gaussian Copulas. In G. Sidorov, A. Hernández Aguirre, and C.A. Reyes García, editors, *Advances in Soft Computing*, volume 6438 of *Lectures Notes in Artificial Intelligence*, pages 104–115. Springer, 2010b. 18, 74
- R. Salinas-Gutiérrez, A. Hernández-Aguirre, and E.R. Villa-Diharce. D-vine EDA: a new Estimation of Distribution Algorithm based on Regular Vines. In *GECCO '10: Proceedings of the 12th annual conference on Genetic and Evolutionary Computation*, pages 359–366, New York, NY, USA, 2010c. ACM. ISBN 978-1-4503-0072-8. 5, 17, 67
- R. Salinas-Gutiérrez, A. Hernández-Aguirre, and E.R. Villa-Diharce. Copula Selection for Graphical Models in Continuous Estimation of Distribution Algorithm. *Computational Statistics*, 2011a. Submitted for revision. 5, 46
- R. Salinas-Gutiérrez, A. Hernández-Aguirre, and E.R. Villa-Diharce. Dependence Trees with Copula Selection for Continuous Estimation of Distribution Algorithms. In *GECCO '11: Proceedings of the 13th annual conference on Genetic and Evolutionary Computation*, pages 585–592. ACM, 2011b. ISBN 978-1-4503-0557-0. Estimation of Distribution Algorithms Track Papers. 5, 17, 46
- R. Santana-Hermida. *Modelación probabilística basada en modelos gráficos no dirigidos en Algoritmos Evolutivos con Estimación de Distribuciones*. PhD thesis, Instituto de Cibernética, Matemáticas y Física, La Habana, Cuba, February 2004. In Spanish. 4, 33

- C. Schölzel and P. Friederichs. Multivariate non-normally distributed random variables in climate research – introduction to the copula approach. *Non-linear Processes in Geophysics*, 15(5):761–772, 2008. ISSN 1023-5809. URL <http://www.nonlin-processes-geophys.net/15/761/2008/>. 2, 17
- M. Sebag and A. Ducoulombier. Extending Population-Based Incremental Learning to Continuous Search Spaces. In A.E. Eiben, T. Bäck, M. Schoenauer, and H.P. Schwefel, editors, *Parallel Problem Solving from Nature – PPSN V*, volume 1498 of *Lecture Notes in Computer Science*, pages 418–427. Springer Berlin / Heidelberg, 1998. 9
- I. Servet, L. Trave-Massuyes, and D. Stern. Telephone Network Traffic Overloading Diagnosis and Evolutionary Computation Techniques. In J.K. Hao, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, editors, *Artificial Evolution. Third European Conference AE '97*, volume 1363 of *Lecture Notes in Computer Science*, pages 137–144. Springer Berlin / Heidelberg, 1997. 9
- S.K. Shakya. *DEUM: A framework for an Estimation of Distribution Algorithm based on Markov Random Fields*. PhD thesis, The Robert Gordon University, Aberdeen, United Kingdom, April 2006. 4, 10, 33
- A. Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris*, 8:229–231, 1959. 17
- M. Soto, A. Ochoa, S. Acid, and L.M. de Campos. Introducing the polytree approximation of distribution algorithm. In A. Ochoa, M. Soto, and R. Santana, editors, *Second International Symposium on Artificial Intelligence. Adaptive Systems. CIMAF99*, pages 360–367, La Habana, 1999. Academia. ISBN 959-02-0241-1. 4, 14, 33
- M.R. Soto-Ortiz. *Un estudio sobre los Algoritmos Evolutivos con Estimación de Distribuciones basados en poliárboles y su costo de evaluación*. PhD thesis, Instituto de Cibernética, Matemáticas y Física, La Habana, Cuba, July 2003. In Spanish. 4, 14
- Y. Stitou, N. Lasmar, and Y. Berthoumieu. Copulas based multivariate gamma modeling for texture classification. In *ICASSP '09: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1045–1048, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-1-4244-2353-8. doi: <http://dx.doi.org/10.1109/ICASSP.2009.4959766>. 17
- P.K. Trivedi and D.M. Zimmer. *Copula Modeling: An Introduction for Practitioners*, volume 1 of *Foundations and Trends® in Econometrics*. Now Publishers, 2007. 2, 17, 18, 22, 42
- G. Venter, J. Barnett, R. Kreps, and J. Major. Multivariate Copulas for Financial Modeling. *Variance*, 1(1):103–119, 2007. 17

- L. Wang, X. Guo, J. Zeng, and Y. Hong. Using Gumbel Copula and Empirical Marginal Distribution in Estimation of Distribution Algorithm. In *Third International Workshop on Advanced Computational Intelligence, IWACI 2010*, pages 583–587. IEEE, August 2010a. 5
- L. Wang, J. Zeng, Y. Hong, and X. Guo. Copula Estimation of Distribution Algorithm Sampling from Clayton Copula. *Journal of Computational Information Systems*, 6(7):2431–2440, 2010b. 5
- L.F. Wang and J.C. Zeng. Estimation of Distribution Algorithm Based on Copula Theory. In Y.P. Chen, editor, *Exploitation of Linkage Learning in Evolutionary Algorithms*, volume 3 of *Adaptation, Learning, and Optimization*, pages 139–162. Springer, 2010. 5
- L.F. Wang, J.C. Zeng, and Y. Hong. Estimation of Distribution Algorithm Based on Copula Theory. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1057–1063. IEEE Press, May 2009a. 5
- L.F. Wang, J.C. Zeng, and Y. Hong. Estimation of Distribution Algorithm Based on Archimedean Copulas. In *GEC '09: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, pages 993–996, New York, NY, USA, June 2009b. ACM. ISBN 978-1-60558-326-6. doi: <http://doi.acm.org/10.1145/1543834.1543991>. 5
- L.F. Wang, Y.C. Wang, J.C. Zeng, and Y. Hong. An Estimation of Distribution Algorithm Based on Clayton Copula and Empirical Margins. In *Life System Modeling and Intelligent Computing*, pages 82–88. Springer, September 2010c. 5
- Gregor Weiß. Copula parameter estimation by maximum-likelihood and minimum-distance estimators: a simulation study. *Computational Statistics*, 26(1):31–54, 2011. 24
- J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. John Wiley & Sons, 1990. 34

Bibliography

Appendix A

Selected copula functions

There are many copula functions. This appendix shows a brief description of the copula functions selected for this thesis.

A.1 Ali-Mikhail-Haq

Distribution function

$$C(u, v) = \frac{uv}{1 - \theta(1-u)(1-v)}$$

Conditional distribution function

$$\frac{\partial C}{\partial u} = \frac{v(1 - \theta(1-v))}{(1 - \theta(1-u)(1-v))^2}$$

Density function

$$c(u, v) = \frac{1 + \theta(u + v + uv - 2) - \theta^2(u + v - uv - 1)}{(1 - \theta(1-u)(1-v))^3}$$

Dependence parameter

$$\theta \in [-1, 1)$$

Kendall's tau

$$\tau = \left(\frac{3\theta - 2}{3\theta} \right) - \frac{2}{3} \left(1 - \frac{1}{\theta} \right)^2 \ln(1 - \theta)$$

Copula class: Archimedean. The generator function is shown in Table 3.1.

A.2 Clayton

Distribution function

$$C(u, v) = \max \left\{ (u^{-\theta} + v^{-\theta} - 1)^{-1/\theta}, 0 \right\}$$

Conditional distribution function

$$\frac{\partial C}{\partial u} = u^{-\theta-1} (u^{-\theta} + v^{-\theta} - 1)^{-\frac{1}{\theta}-1}$$

Density function

$$c(u, v) = (1 + \theta) (uv)^{-\theta-1} (u^{-\theta} + v^{-\theta} - 1)^{-2-1/\theta}$$

Dependence parameter

$$\theta \in [-1, \infty) \setminus \{0\}$$

Kendall's tau

$$\tau = \frac{\theta}{\theta + 2}$$

Copula class: Archimedean. The generator function is shown in Table 3.1.

A.3 Farlie-Gumbel-Morgenstern

Distribution function

$$C(u, v) = uv (1 + \theta(1 - u)(1 - v))$$

Conditional distribution function

$$\frac{\partial C}{\partial u} = v^2 (\theta(2u - 1)) + v (1 + \theta(1 - 2u))$$

Density function

$$c(u, v) = 1 + \theta(1 - 2u)(1 - 2v)$$

Dependence parameter

$$\theta \in [-1, 1]$$

Kendall's tau

$$\tau = \frac{2}{9}\theta$$

Copula class: Nor Archimedean, nor elliptical.

A.4 Frank

Distribution function

$$C(u, v) = -\frac{1}{\theta} \ln \left(1 + \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1} \right)$$

Conditional distribution function

$$\frac{\partial C}{\partial u} = \frac{e^{-\theta u}(e^{-\theta v} - 1)}{(e^{-\theta u} - 1)(e^{-\theta v} - 1) + (e^{-\theta} - 1)}$$

Density function

$$c(u, v) = \frac{-\theta(e^{-\theta} - 1)e^{-\theta(u+v)}}{((e^{-\theta u} - 1)(e^{-\theta v} - 1) + (e^{-\theta} - 1))^2}$$

Dependence parameter

$$\theta \in (-\infty, \infty) \setminus \{0\}$$

Kendall's tau

$$\tau = 1 - \frac{4}{\theta} \left[1 - \frac{1}{\theta} \int_0^\theta \frac{t}{e^t - 1} dt \right]$$

Copula class: Archimedean. The generator function is shown in Table 3.1.

A.5 Gaussian

Distribution function

$$C(u, v) = \int_{-\infty}^{\Phi^{-1}(u)} \int_{-\infty}^{\Phi^{-1}(v)} \frac{e^{-\frac{1}{2}t'\Sigma^{-1}t}}{2\pi|\Sigma|^{1/2}} dt_1 dt_2$$

where Σ is a correlation matrix with $\Sigma_{12} = \theta$

Density function

$$c(u, v) = (1 - \theta^2)^{1/2} \exp \left(-\frac{(x^2 + y^2 - 2\theta xy)}{2(1 - \theta^2)} + \frac{(x^2 + y^2)}{2} \right)$$

where $x = \Phi^{-1}(u)$ and $y = \Phi^{-1}(v)$

Dependence parameter

$$\theta \in (-1, 1)$$

Kendall's tau

$$\tau = \frac{2}{\pi} \sin^{-1}(\theta)$$

Copula class: Elliptical.

A.6 Gumbel

Distribution function

$$C(u, v) = \exp\left(-(\tilde{u}^\theta + \tilde{v}^\theta)^{1/\theta}\right)$$

where $\tilde{u} = -\ln(u)$ and $\tilde{v} = -\ln(v)$

Conditional distribution function

$$\frac{\partial C}{\partial u} = \left(\frac{\ln u}{\ln C(u, v)}\right)^{\theta-1} \frac{C(u, v)}{u}$$

Density function

$$c(u, v) = \frac{C(u, v)}{uv} \frac{(\tilde{u}\tilde{v})^{\theta-1}}{(\tilde{u}^\theta + \tilde{v}^\theta)^{2-1/\theta}} \left((\tilde{u}^\theta + \tilde{v}^\theta)^{1/\theta} + \theta - 1\right)$$

where $\tilde{u} = -\ln(u)$ and $\tilde{v} = -\ln(v)$

Dependence parameter

$$\theta \in [1, \infty)$$

Kendall's tau

$$\tau = 1 - \frac{1}{\theta}$$

Copula class: Archimedean. The generator function is shown in Table 3.1.

Appendix B

Benchmark functions

This appendix shows a brief description of the test functions selected for this thesis.

B.1 Ackley

Mathematical definition

$$g(\mathbf{x}) = -20 \cdot \exp\left(-0.2 \sqrt{\frac{1}{d} \cdot \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \cdot \sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + \exp(1)$$

Search domain

$$\mathbf{x} \in [-10, 10]^d$$

Global minimum

$$g(\mathbf{0}) = 0$$

Properties: Multimodal, non-separable.

B.2 Cigar

Mathematical definition

$$g(\mathbf{x}) = x_1^2 + \sum_{i=2}^d 10^6 x_i^2$$

Search domain

$$\mathbf{x} \in [-10, 5]^d$$

Global minimum

$$g(\mathbf{0}) = 0$$

Properties: Unimodal, separable.

B.3 Cigar Tablet

Mathematical definition

$$g(\mathbf{x}) = x_1^2 + \sum_{i=2}^{d-1} 10^4 x_i^2 + 10^8 x_d^2$$

Search domain

$$\mathbf{x} \in [-10, 5]^d$$

Global minimum

$$g(\mathbf{0}) = 0$$

Properties: Unimodal, separable.

B.4 Ellipsoid

Mathematical definition

$$g(\mathbf{x}) = \sum_{i=1}^d 10^{6 \frac{i-1}{d-1}} x_i^2$$

Search domain

$$\mathbf{x} \in [-10, 5]^d$$

Global minimum

$$g(\mathbf{0}) = 0$$

Properties: Unimodal, separable.

B.5 Griewangk

Mathematical definition

$$g(\mathbf{x}) = 1 + \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

Search domain

$$\mathbf{x} \in [-600, 600]^d$$

Global minimum

$$g(\mathbf{0}) = 0$$

Properties: Multimodal, non-separable.

B.6 Rastrigin

Mathematical definition

$$g(\mathbf{x}) = \sum_{i=1}^d (x_i^2 - 10\cos(2\pi x_i) + 10)$$

Search domain

$$\mathbf{x} \in [-5.12, 5.12]^d$$

Global minimum

$$g(\mathbf{0}) = 0$$

Properties: Multimodal, separable.

B.7 Rosenbrock

Mathematical definition

$$g(\mathbf{x}) = \sum_{i=1}^{d-1} [100 \cdot (x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$$

Search domain

$$\mathbf{x} \in [-10, 10]^d$$

Global minimum

$$g(\mathbf{1}) = 0$$

Properties: Unimodal, non-separable.

B.8 Schwefel 1.2 (Quadric)

Mathematical definition

$$g(\mathbf{x}) = \sum_{i=1}^d \left(\sum_{j=1}^i x_j \right)^2$$

Search domain

$$\mathbf{x} \in [-40, 60]^d$$

Global minimum

$$g(\mathbf{0}) = 0$$

Properties: Unimodal, non-separable.

B.9 Sphere Model

Mathematical definition

$$g(\mathbf{x}) = \sum_{i=1}^d x_i^2$$

Search domain

$$\mathbf{x} \in [-600, 600]^d$$

Global minimum

$$g(\mathbf{0}) = 0$$

Properties: Unimodal, separable.

B.10 Trid

Mathematical definition

$$g(\mathbf{x}) = \sum_{i=1}^d (x_i - 1)^2 - \sum_{i=2}^d x_i x_{i-1}$$

Search domain

$$\mathbf{x} \in [-d^2, d^2]^d$$

Global minimum

$$g(\mathbf{x}) = \frac{-d(d+4)(d-1)}{6}$$

Properties: Unimodal, non-separable.

B.11 Two Axes

Mathematical definition

$$g(\mathbf{x}) = \sum_{i=1}^{\lfloor d/2 \rfloor} 10^6 x_i^2 + \sum_{i=\lfloor d/2 \rfloor}^d x_i^2$$

Search domain

$$\mathbf{x} \in [-10, 5]^d$$

Global minimum

$$g(\mathbf{0}) = 0$$

Properties: Unimodal, separable.

B.12 Zakharov

Mathematical definition

$$g(\mathbf{x}) = \sum_{i=1}^d x_i^2 + \left(\sum_{i=1}^d 0.5ix_i \right)^2 + \left(\sum_{i=1}^d 0.5ix_i \right)^4$$

Search domain

$$\mathbf{x} \in [-5, 10]^d$$

Global minimum

$$g(\mathbf{0}) = 0$$

Properties: Unimodal, non-separable.

Acknowledgements

During the last four years I have received the support of several institutions and many people in order to do my doctoral studies. So, I would like to express my sincere gratitude to them for their invaluable help.

First and foremost, I would like to express my gratitude and appreciation to my advisor Dr. Arturo Hernández Aguirre for his continuous support, encouragement, guidance and constant patience during the preparation of my thesis. A special thanks also to my co-advisor Dr. Enrique Raúl Villa Diharce, who has played a supportive role through statistical advising.

I am grateful to Dr. Elva Díaz Díaz, Dr. Jean Bernard Hayet, Dr. Rogelio Ramos Quiroga, Dr. Salvador Ruiz Correa, and again Dr. Enrique Raúl Villa Diharce for accepting being part of the thesis committee, and for their constructive comments and suggestions during the revisions of the thesis and the oral defense.

I acknowledge financial support from the Consejo Nacional de Ciencia y Tecnología (CONACYT) through a scholarship to pursue graduate studies at the Centro de Investigación en Matemáticas. This scholarship was awarded during the four years of my doctoral studies.

I had the opportunity of attending the Mexican International Conference on Artificial Intelligence (MICA 2009, MICA 2010) and the Genetic and Evolutionary Computation Conference (GECCO 2010, GECCO 2011). My participations in these conferences in order to present my research works were possible thanks to the financial support of the research project of my advisor, the Centro de Investigación en Matemáticas, and the student travel grants awarded by the Association for Computing Machinery Special Interest Group on Genetic and Evolutionary Computation (ACM SIGEVO).

I also had the wonderful experience of having participated in the workshop “Mathematical Modeling in Industry XIV” thanks to the financial support and the opportunity given by the Institute for Mathematics and its Applications (IMA), the Pacific Institute for the Mathematical Sciences (PIMS) and the Centro de Investigación en Matemáticas (CIMAT).

Special gratitude is given to the Centro de Investigación en Matemáticas and its people: cubicle mates, administrative people, classmates and professors. I can proudly state it was a great pleasure to do my doctoral studies in such an interdisciplinary and stimulating environment. A special mention goes out to the English teacher of the Centro de Investigación en Matemáticas, Stephanie Dunbar, for her invaluable support in order to improve my presentations and my written papers. Thanks to Joaquín Peña for showing me how to improve the quality of my documents by using \LaTeX .

It would not have been possible to do my doctoral studies without the support of the Universidad Autónoma de Aguascalientes along with the Programa de Mejoramiento del Profesorado (PROMEP). From the university authorities, Rafael Urzúa Macías, Francisco Javier Álvarez Rodríguez, and Javier Bech Vertti, I received the academic permission for pursuing graduate studies without losing my job, whereas from the PROMEP, I received a complementary financial

Acknowledgements

support. Thanks to these supports, I will return to my job at the Universidad Autónoma de Aguascalientes with a strong academic preparation for teaching and doing research.

During the last year I met wonderful people in Guanajuato. I am grateful to Oscar Dalmau Cedeño, Roberto Cruz Oropesa, Isabel Tatiana Rodríguez Esnard and Yudeysi Ysada Borrego. We shared music, excellent food, experiences and good times.

My relatives and friends have been an endless support during the years I have spent in Guanajuato. My brother Edgar Iván and my mom Martha deserve special thanks for giving my wife and my children all the assistance they needed during the last year in Aguascalientes.

Finally, I would like to thank the support of my wife Leticia and my children Sofía Carolina and Rogelio Emiliano. They have been a continuous source of inspiration and love. I dedicate this thesis to them.

Guanajuato, August 2011.
Rogelio Salinas

About the author

Biographical Sketch

Rogelio Salinas Gutiérrez was born on September 11, 1975 in Aguascalientes, México. In 1996 he started his studies in applied mathematics at the Universidad Autónoma de Aguascalientes and received his bachelor's degree 5 years later.

In 2001 Rogelio moved to Guanajuato, México where he followed a master of science program at the Centro de Investigación en Matemáticas. In July 2003 Rogelio defended his master thesis “Cálculo de Frecuencias Naturales en Vigas Elásticas, Un Estudio Comparativo” (Computation of natural frequencies for Elastic Beams, a Comparative Approach) and obtained the master's degree in Computer Science and Industrial Mathematics. His master work was performed under the supervision of Dr. Miguel Ángel Moreles Vázquez and Dr. Salvador Botello Rionda.

Directly after obtaining his master's degree, Rogelio returned to Aguascalientes and during the following four years he was dedicated to teaching and was involved in the local committee of the mathematical olympiad. He worked as an university teacher of mathematics and statistics at the Universidad Autónoma de Aguascalientes and the Universidad Panamericana, campus Bonaterra. In several times he was invited for giving talks in academic events. The topics of his presentations were about optimization, numerical methods, mathematics, statistics and their applications. Furthermore, he got a Diploma in Teaching for Distance Education in 2006 from the Universidad Autónoma de Aguascalientes.

In August 2007 Rogelio was accepted for doing doctoral studies at the Centro de Investigación en Matemáticas under the supervision of Dr. Arturo Hernández Aguirre and Dr. Enrique Raúl Villa Diharce. In his research work he has been involved in the study and the use of copula functions into evolutionary computation.

Currently Rogelio is a full time professor in the Statistics Department at the Universidad Autónoma de Aguascalientes. Rogelio is married and has two children.

Publications during doctoral studies

Related to this thesis

Rogelio Salinas-Gutiérrez, Arturo Hernández-Aguirre, and Enrique R. Villa-Diharce. Copula Selection for Graphical Models in Continuous Estimation of Distribution Algorithms. *Computational Statistics*, 2011. Submitted for revision.

Rogelio Salinas-Gutiérrez, Arturo Hernández-Aguirre, and Enrique R. Villa-Diharce. Estimation of Distribution Algorithms based on Copula Functions.

In *GECCO '11: Proceedings of the 13th annual conference on Genetic and Evolutionary Computation*, pages 795–798. ACM, 2011. Accepted for the Graduate Students Workshop.

Rogelio Salinas-Gutiérrez, Arturo Hernández-Aguirre, and Enrique R. Villa-Diharce. Dependence Trees with Copula Selection for Continuous Estimation of Distribution Algorithms. In *GECCO '11: Proceedings of the 13th annual conference on Genetic and Evolutionary Computation*, pages 585–592. ACM, 2011. Accepted for the EDAs Track Papers.

Rogelio Salinas-Gutiérrez, Arturo Hernández-Aguirre, and Enrique R. Villa-Diharce. D-vine EDA: a new Estimation of Distribution Algorithm based on Regular Vines. In *GECCO '10: Proceedings of the 12th annual conference on Genetic and Evolutionary Computation*, pages 359–366. ACM, 2010.

Rogelio Salinas-Gutiérrez, Arturo Hernández-Aguirre, and Enrique R. Villa-Diharce. Using Copulas in Estimation of Distribution Algorithms. In A. Hernández Aguirre, R. Monroy Borja, and C.A. Reyes García, editors, *MICAI 2009: Advances in Artificial Intelligence*, volume 5845 of *Lecture Notes in Artificial Intelligence*, pages 658–668. Springer, 2009.

Rogelio Salinas Gutiérrez, Arturo Hernández Aguirre, and Enrique Raúl Villa Diharce. Una extensión del algoritmo MIMIC mediante Cópulas. *Revista Electrónica Nova Scientia*, 2(3):1–13, 2009.
Available at http://nova_scientia.delasalle.edu.mx/numero_3/index.html

Partially related to this thesis

Rogelio Salinas-Gutiérrez, Arturo Hernández-Aguirre, Mariano J.J. Rivera-Meraz, and Enrique R. Villa-Diharce. Supervised Probabilistic Classification Based on Gaussian Copulas. In G. Sidorov, A. Hernández Aguirre, and C.A. Reyes García, editors, *Advances in Soft Computing*, volume 6438 of *Lecture Notes in Artificial Intelligence*, pages 104–115. Springer, 2010.

Rogelio Salinas-Gutiérrez, Arturo Hernández-Aguirre, Mariano J.J. Rivera-Meraz, and Enrique R. Villa-Diharce. Using Gaussian Copulas in Supervised Probabilistic Classification. In O. Castillo, J. Kacprzyk, and W. Pedrycz, editors, *Soft Computing for Intelligent Control and Mobile Robotics*, volume 318 of *Studies in Computational Intelligence*, pages 355–372. Springer. 2010.

Not related to this thesis

Michael Hofmeister, Sean Ahmad Colbert-Kelly, Kirill Levin, Huijuan Li, Ignacio Rozada, Rogelio Salinas Gutiérrez, and Benjamin Sánchez Lengeling. Pro-

duction planning for water supply networks. IMA Preprint Series # 2334, University of Minnesota, September 2010.

Available at <http://www.ima.umn.edu/preprints/sep2010/sep2010.html>

Arturo Hernández Aguirre, Sergio Ivvan Valdez Peña, Ángel Eduardo Muñoz Zavala, Giovanni Lizárraga Lizárraga, and Rogelio Salinas Gutiérrez, editors. *Avances en Computación Evolutiva. Memorias del IV Congreso Mexicano de Computación Evolutiva COMCEV 2008*. Centro de Investigación en Matemáticas, 2008.

Publications before doctoral studies

Miguel Angel Moreles, Salvador Botello, and Rogelio Salinas. A root-finding technique to compute eigenfrequencies for elastic beams. *Journal of Sound and Vibration*, 284(3-5):1119–1129, 2005.

Miguel Ángel Moreles, Salvador Botello, and Rogelio Salinas. Cálculo de frecuencias naturales para vigas elásticas con efecto de cortante e inercia rotacional. Caso empotrado-articulado. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, 21(1):3–21, 2005.

Miguel A. Moreles Vázquez, Salvador Botello Rionda, and Rogelio Salinas Gutiérrez. Un método asintótico para calcular las frecuencias naturales en vigas elásticas. In S. Gallegos, I. Herrera, S. Botello, F. Zárate, and G. Ayala, editors, *Memorias del III Congreso Internacional sobre Métodos Numéricos en Ingeniería y Ciencias Aplicadas*, ITESM-CIMNE, 2004.

Available at

<http://congress.cimne.upc.es/mtv2004/memorias/html/III Congreso.htm>

Miguel Ángel Moreles, Salvador Botello, and Rogelio Salinas. *Computation of eigenfrequencies for elastic beams, a comparative approach*. Editorial Aula CIMNE-UGTO, 2004.

Rogelio Salinas Gutiérrez. Cálculo de Frecuencias Naturales en Vigas Elásticas, Un Estudio Comparativo. Master's thesis, Centro de Investigación en Matemáticas, Guanajuato, México, July 2003.

Available at <http://www.cimat.mx/index.php?m=382>

Miguel Angel Moreles, Salvador Botello Salvador, and Rogelio Salinas. Computation of Eigenfrequencies for Elastic Beams, a Comparative Approach. Comunicación Técnica del CIMAT I-03-09/25-04-2003, Centro de Investigación en Matemáticas, 2003.

Available at <http://www.cimat.mx/index.php?m=279>

About the author
