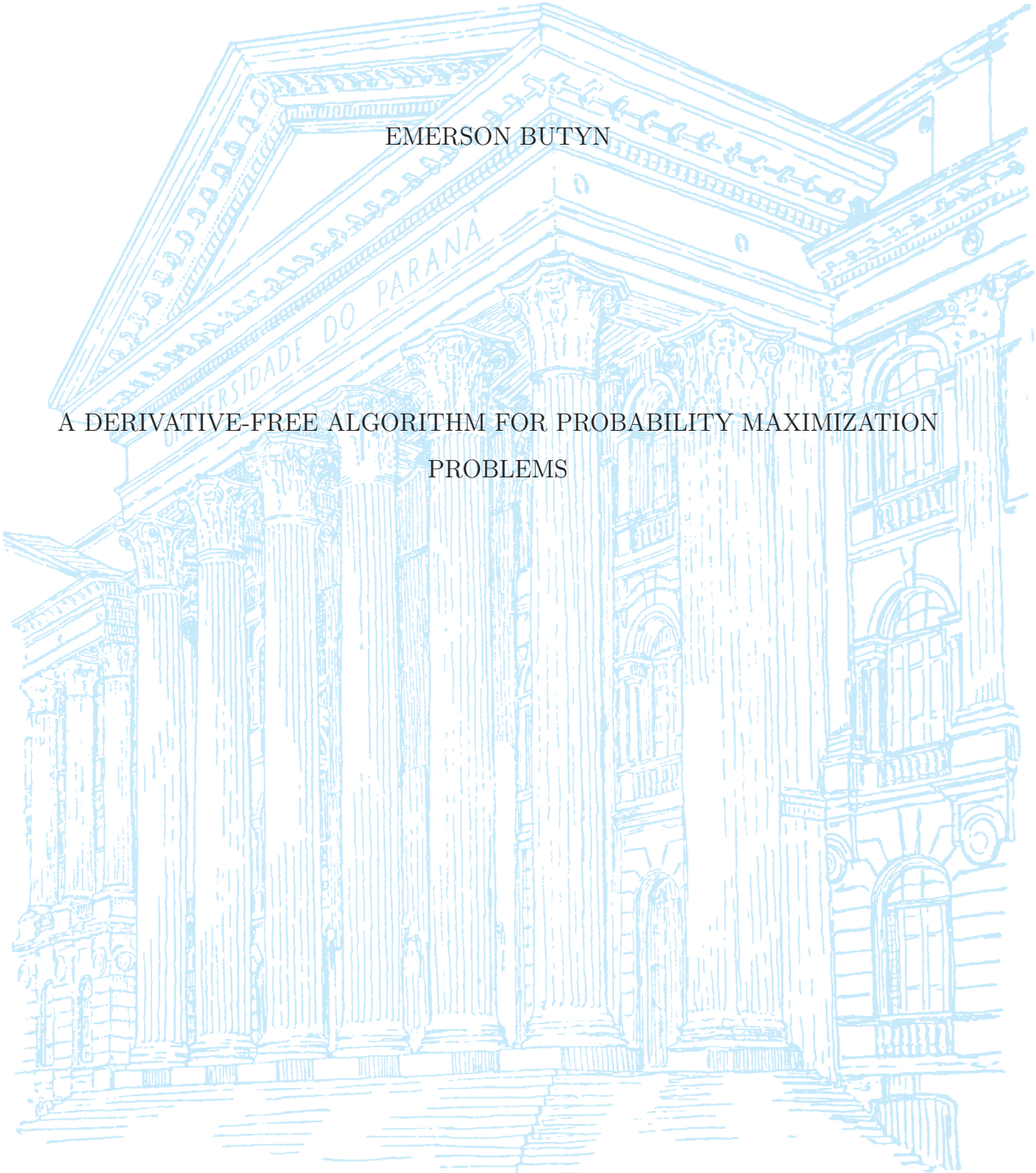
UNIVERSIDADE FEDERAL DO PARANÁ

EMERSON BUTYN



A DERIVATIVE-FREE ALGORITHM FOR PROBABILITY MAXIMIZATION PROBLEMS

CURITIBA

2021

EMERSON BUTYN

A DERIVATIVE-FREE ALGORITHM FOR PROBABILITY MAXIMIZATION PROBLEMS

Tese apresentada ao Programa de Pós Graduação em Matemática, Setor de Ciências Exatas, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Doutor em Matemática.

Orientadora: Dra. Elizabeth Wegner Karas
UFPR - Brasil

Coorientador: D. Habil. Welington Luis de Oliveira - MINES ParisTech - France

CURITIBA

2021

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA CIÊNCIA E TECNOLOGIA

Emerson Butyn
A derivative-free algorithm for probability maximization problems / Emerson Butyn – Curitiba, 2021.
1 recurso on-line : PDF.

Tese (Doutorado) – Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Matemática.
Orientadora: Dra. Elizabeth Wegner Karas (UFPR – Brasil)
Coorientador: D. Habil. Welington Luis de Oliveira (MINES ParisTech - France)

1. Programação não-linear. 2. Otimização matemática. 3. Programação estocástica. I. Karas, Elizabeth Wegner. II. Oliveira, Welington Luis de. III. Universidade Federal do Paraná. Programa de Pós-Graduação em Matemática. IV. Título.

Bibliotecária: Roseny Rivelini Morciani CRB-9/1585

MINISTÉRIO DA EDUCAÇÃO
SETOR DE CIENCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO MATEMÁTICA - 40001016041P1

ATA Nº042

ATA DE SESSÃO PÚBLICA DE DEFESA DE DOUTORADO PARA A OBTENÇÃO DO GRAU DE DOUTOR EM MATEMÁTICA

No dia vinte e dois de novembro de dois mil e vinte e um às 09:00 horas, na sala https://meet.jit.si/DefenseEmersonButynThesis_November22_9BR, virtual, foram instaladas as atividades pertinentes ao rito de defesa de tese do doutorando **EMERSON BUTYN**, intitulada**: A derivative-free algorithm for probability maximization problems**. A Banca Examinadora, designada pelo Colegiado do Programa de Pós-Graduação MATEMÁTICA da Universidade Federal do Paraná, foi constituída pelos seguintes Membros: WELINGTON LUIS DE OLIVEIRA (ÉCOLE NATIONALE SUPÉRIEURE DES MINES DE PARIS), JOSÉ MARIO MARTÍNEZ (UNIVERSIDADE ESTADUAL DE CAMPINAS), MAEL SACHINE (UNIVERSIDADE FEDERAL DO PARANÁ), FRANCISCO NOGUEIRA CALMON SOBRAL (UNIVERSIDADE ESTADUAL DE MARINGÁ), WIM VAN ACKOOIJ (ÉLECTRICITÉ DE FRANCE). A presidência iniciou os ritos definidos pelo Colegiado do Programa e, após exarados os pareceres dos membros do comitê examinador e da respectiva contra argumentação, ocorreu a leitura do parecer final da banca examinadora, que decidiu pela APROVAÇÃO. Este resultado deverá ser homologado pelo Colegiado do programa, mediante o atendimento de todas as indicações e correções solicitadas pela banca dentro dos prazos regimentais definidos pelo programa. A outorga de título de doutor está condicionada ao atendimento de todos os requisitos e prazos determinados no regimento do Programa de Pós-Graduação. Nada mais havendo a tratar a presidência deu por encerrada a sessão, da qual eu, WELINGTON LUIS DE OLIVEIRA, lavrei a presente ata, que vai assinada por mim e pelos demais membros da Comissão Examinadora.

CURITIBA, 22 de Novembro de 2021.

Assinatura Eletrônica
22/11/2021 12:27:53.0
WELINGTON LUIS DE OLIVEIRA
Presidente da Banca Examinadora

Assinatura Eletrônica
23/11/2021 07:53:03.0
JOSÉ MARIO MARTÍNEZ
Avaliador Externo (UNIVERSIDADE ESTADUAL DE CAMPINAS)

Assinatura Eletrônica
22/11/2021 12:25:58.0
MAEL SACHINE
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
22/11/2021 15:15:30.0
FRANCISCO NOGUEIRA CALMON SOBRAL
Avaliador Externo (UNIVERSIDADE ESTADUAL DE MARINGÁ)

Assinatura Eletrônica
22/11/2021 12:28:04.0
WIM VAN ACKOOIJ
Avaliador Externo (ÉLECTRICITÉ DE FRANCE)

MINISTÉRIO DA EDUCAÇÃO
SETOR DE CIENCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO MATEMÁTICA - 40001016041P1

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação MATEMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **EMERSON BUTYN** intitulada: **A derivative-free algorithm for probability maximization problems**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 22 de Novembro de 2021.

Assinatura Eletrônica
22/11/2021 12:27:53.0
WELINGTON LUIS DE OLIVEIRA
Presidente da Banca Examinadora

Assinatura Eletrônica
23/11/2021 07:53:03.0
JOSÉ MARIO MARTÍNEZ
Avaliador Externo (UNIVERSIDADE ESTADUAL DE CAMPINAS)

Assinatura Eletrônica
22/11/2021 12:25:58.0
MAEL SACHINE
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
22/11/2021 15:15:30.0
FRANCISCO NOGUEIRA CALMON SOBRAL
Avaliador Externo (UNIVERSIDADE ESTADUAL DE MARINGÁ)

Assinatura Eletrônica
22/11/2021 12:28:04.0
WIM VAN ACKOOIJ
Avaliador Externo (ÉLECTRICITÉ DE FRANCE)

ACKNOWLEDGEMENT

Firstly, I would like to thank my wife Tatiane for her patience, love and understanding of my PhD journey, and my family, for supporting me during these last few years.

I am really grateful to my advisors Elizabeth Wegner Karas and Welington de Oliveira for all the the moments we have been chating, working and making progress of our research in online meetings. I am so pleased for their support, guidance and teachings.

I would like to thank my thesis comitee, Wim van Ackooij (EDF - France), Francisco Sobral (UEM), José Mario Martínez (UNICAMP) and Mael Sachine (UFPR) for their comments and suggestions, which provided a valuable contribution to this work.

I can not forget to mention and thank my friends that studied with me in the first two years of doctorate in many disciplines, where supporting each other was what helped me get here.

Lastly, thanks to CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) which financially supported the development of this work.

RESUMO

Nesta tese propomos um algoritmo de região de confiança sem derivadas para problemas de maximização de probabilidade. Assumimos que a função de probabilidade é continuamente diferenciável com gradiente Lipschitz contínuo, mas nenhuma derivada está disponível. O algoritmo explora a estrutura particular da função objetivo de probabilidade por meio de modelos baseados em cópulas. Sob hipóteses razoáveis, a convergência global do algoritmo é analisada. Provamos que todos os pontos de acumulação da sequência gerada pelo algoritmo são estacionários. A proposta é validada através de experimentos numéricos na resolução de problemas acadêmicos e industriais.

Keywords: Programação não linear, Problema de maximização de probabilidade, Programação estocástica, Otimização sem derivadas.

ABSTRACT

In this thesis, we propose a derivative-free trust-region algorithm for probability maximization problems. We assume that the probability function is continuously differentiable with Lipschitz continuous gradient, but no derivatives are available. The algorithm explores the particular structure of the probability objective function through models based on copulæ. Under reasonable assumptions, the global convergence of the algorithm is analyzed. In fact, we prove that all accumulation points of the sequence generated by the algorithm are stationary. The proposed approach is validated by encouraging numerical results on academic and industrial problems.

Keywords: Nonlinear programming, Probability maximization problems, Stochastic programming, Derivative-free optimization.

Contents

1 Introduction **11**

2 Probability distribution functions **21**

2.1 Basic concepts 21

2.2 Continuity and differentiability of distribution functions 28

2.2.1 Continuity 28

2.2.2 Differentiability 30

2.3 Copulæ 45

2.3.1 Definition and properties 46

2.3.2 Families of copulæ 56

3 Trust-region algorithm with copula-based models **65**

3.1 Copula-based model 65

3.1.1 Application of Sklar's theorem 66

3.1.2 The model 67

3.1.3 The algorithm 68

3.2 Convergence analysis 71

3.2.1 Assumptions A1-A3: what do they yield? 73

3.2.2 Global convergence of the algorithm 75

3.2.3 Ensuring Assumption A3 81

3.3 Efficiency condition for the subproblem's stationary point 84

4 Numerical experiments **87**

4.1 Nonlinear continuous problems 88

4.1.1 Solvers 88

4.1.2 Test problems and numerical experiments 91

4.2 Mixed-Integer Nonlinear Programming problems 102

4.2.1 Solvers 103

4.2.2 Test problem and numerical results 104

5 Conclusion **109**

References **111**

Chapter 1

Introduction

Many real-life situations can be modeled as optimization problems, in which practitioners and researchers wish to minimize or maximize a real function over a set of constraints. Depending on whether data is known or random, the underlying optimization problem can be classified as deterministic or stochastic. In the latter case, very often, one needs to cope with a random inequality system of the form

$$\xi \leq g(x), \tag{1.1}$$

where ξ is a m-dimensional random vector defined on the probability space $(\Xi, \mathcal{X}, \mathbb{P})$ with continuous probability measure $\mathbb{P}$, $g : \mathcal{O} \to \mathbb{R}^m$ is a function of class $\mathcal{C}^1$ with Lipschitz continuous gradient defined in an open set $\mathcal{O} \subset \mathbb{R}^n$.

Throughout this work, we assume that the continuous probability distribution of ξ is known and independent of the decision vector x. Furthermore, we assume that x belongs to a nonempty compact convex set (typically a polyhedron) $X \subset \mathbb{R}^n$ such that $X \subset \mathcal{O}$. This thesis is dedicated to the stochastic programming problem of finding a point x in X satisfying the random inequality system (1.1) with the highest possible probability. More specifically, we are interested in *Probability Maximization Problems* (PMPs) of the form

$$\max_{x \in X} \varphi(x), \quad \text{with} \quad \varphi(x) := \mathbb{P}\left[\xi \leq g(x)\right]. \tag{1.2}$$

Many applications from finance and engineering can be formulated as PMPs. For instance, in capacity expansion planning problems under uncertainty, one wishes to expand production capacity with limited resources and capital. Given a budget, one seeks to make a decision on expansion so that physical and monetary constraints are satisfied. The latter constraints can be abstractly represented by the set X. Among the infinite number of possible expansion plans in X, the decision maker searches for a plan of action satisfying a random demand ξ as much as possible, i.e., a decision that maximizes the probability function φ.

Another application of interest is the management of hydro-thermal power systems, where one seeks to produce enough electricity, at the minimal costs, by combining hydro and thermal power generations. Since water has multiple usages, reservoir levels should remain within predefined bounds (for irrigation or tourism reasons). The random nature of the water inflows makes this task difficult. Typically, this assignment is made via optimization models with probability constraints [105], but a PMP approach is perfectly suitable: the system manager searches for a power generation plan, no more expensive than a predefined cost, that maximizes the probability of keeping the reservoirs' volumes within bounds.

As the latter example indicates, PMPs are closely related to optimization problems with probability constraints, also known as Chance-Constrained Problems (CCPs). As explained in [32, 74], for a given real-valued cost function f, a convex set $\tilde{X}$ and a confidence level $p \in (0, 1)$, the classical CCP

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & \varphi(x) \geq p \\ & x \in \tilde{X}, \end{array} \tag{1.3}$$

can be reformulated as (1.2) by defining $X := \{x \in \tilde{X} : f(x) \leq T\}$, where $T \in \mathbb{R}$ is a predefined cost target. Note that it is possible to choose p and T such as CCP and PMP share a solution. We refer the interested reader to excellent textbooks [77] and [83] for an overview of the theory and methods for CCPs, and to the following works [47, 76, 98,

104] on methodologies and applications of optimization problems involving a probability function. Some of these references deal with probability functions even more general than the one of (1.2): sometimes the mapping g also depends on the random vector. The separable setting in (1.1), i.e., when g does not depend on ξ, is not the most general one but is present in many applications. See for instance [103, 104, 105] for applications in energy management, [48] for a problem in finance and [58] for transportation problems.

Since probability maximization problems are special cases of nonlinear optimization, properties of the probability function φ such as continuity, generalized concavity and differentiability can be useful to choose an algorithm for solving the problem. Such properties have been extensively studied [32, 49, 77, 95] in the last years. In particular, the recent paper [92] offers an overview of the state-of-art of probability functions with perspective in variational analysis, highlighting theoretical and algorithmic aspects of these properties.

In this work, we do not assume that φ satisfies any generalized concavity property, and therefore by "solving" problem (1.2) we mean *computing a stationary point*. However, we assume a bit more than differentiability: the function $\varphi : \mathcal{O} \to [0, 1]$ in (1.2) is continuously differentiable with Lipschitz continuous gradient on X, i.e., there exists a (possibly unknown) finite constant $\kappa_\varphi > 0$ such that, for all $x, y \in X$,

$$\|\nabla\varphi(x) - \nabla\varphi(y)\| \leq \kappa_\varphi \|x - y\|. \tag{1.4}$$

Under the assumption that g is of class $\mathcal{C}^1$ with Lipschitz continuous gradient on $\mathcal{O}$, and $X \subset \mathcal{O}$ is a compact set, the condition (1.4) is satisfied by many important probability distributions such as the multivariate Gaussian with positive definite covariance matrix [77, p. 204] or with singular covariance matrix under some nondegeneracy condition [49, Thm. 4.1], and other distributions satisfying some growth conditions [102, Thm. 3] (see § 4 in the latter paper for an analysis on the log-normal and Student distributions). Furthermore, all probability distributions of class $\mathcal{C}^2$ on X satisfy (1.4) (this is a mere consequence of [64, Lem. 1.2.2] together with the assumptions on g and X).

Despite the recent advances in theory and numerical methods for this class of problems,

dealing with multivariate probability functions remains a challenging task, except for some special cases. The main difficulties arise from the fact that typically there is no analytical expression for these functions. Furthermore, numerical evaluation of probability functions and their gradients with reasonable accuracy is too time-consuming even when the random vector is composed of, say, a few dozen components. We recall that computing $\varphi(x)$ for a given x amounts at evaluating numerically a multidimensional integral, a task that can be accomplished in reasonable CPU times only if precision is not a concern. All one can hope for are efficient tools for numerically approximate $\varphi(x)$. Besides that, computing only functional values of φ is not enough to employ some optimization algorithms, it is also necessary to have access to the gradients of φ, which becomes even more involving:

- Approximating $\nabla\varphi(x)$ by finite-difference formulæ is not advisable because it involves several evaluations of φ around x and requires careful selection of finite-difference parameters. As just mentioned, evaluating φ is time consuming depending on the random vector's dimension;

- Algebraic formulae for the gradient of φ are not always available. When accessible, they may not be computationally implementable or practical due to their high complexity. As summarized in [92, §, 2.2], numerical implementations necessary to obtain partial derivatives of probability functions, as well as verifying that all required assumptions are satisfied, are not generally accessible (see [77, § 6.6.4], [90, Thm. 2.1] and [100]). Furthermore, it appears that compactness of the support set Ξ is often assumed, which turns out to be a restriction in some applications.

- Even when an algebraic formula for $\nabla\varphi$ is available and implementable (e.g., when $\xi \in \mathbb{R}^m$ follows a Gaussian distribution), computing $\nabla\varphi(x)$ for a given x is approximately m times more expensive as evaluating $\varphi(x)$. This is due to the fact that gradient formulæ for certain probability distributions require computing m numerical integrations of dimension $m-1$; see for instance [93, Thm. 2.7.3].

Due to the aforementioned difficulties, optimization methods that do not make use of

derivatives appear as a favorable approach for PMPs. Derivative-free optimization (DFO) algorithms are good choices when the gradient of the objective function is not available, or is too difficult to be evaluated. Although this is the case for problem (1.2), we are not aware of any DFO approach specialized for PMPs. This thesis fills this gap by proposing a DFO trust-region method suitable for problems of the form (1.2). We refer the reader to the textbooks [1, 19], methodological papers [9, 46, 110], recent review [54] and tutorial [45] for an overview on DFO methods.

According to [1, 45], DFO algorithms can be split into two broad categories: *direct-search* and *model-based* methods. In [114], it is also considered the class of *implicit filtering* methods [6, 39], that approximate the derivative of the objective function by simplex gradients, a generalization of finite-difference gradient. In order to decrease the objective function, direct-search methods choose points in specific directions with a predefined step size from the incumbent solution, which is updated whenever an improvement condition is achieved, otherwise a new search step size is considered. There are many direct-search methods in the literature, as Hooke and Jeeves' pattern search [50], Generalized Pattern Search (GPS) [52, 56, 57, 86], Nelder-Mead simplex method [62], Mesh Adaptive Direct Search (MADS) [2, 26], Generating Set Search (GSS) [51]. Although, direct-search methods are popular since they are easy to implement and reliable in practice [114], commonly, they require a large amount of function evaluations and do not fully explore the information available of the objective function, making some of them very slow.

On the other hand, model-based methods explore the underlying properties of the objective function rather than its values by themselves. In this class of methods, the function values are used to construct models which should approximate the objective function in a neighbourhood, called trust region, of the current point. Furthermore, to be useful, optimizing the model within this neighborhood has to be significantly easier than solving the original problem. In DFO methods, the models are constructed without any first-order information, by means of polynomial interpolation or regression [19] or by any other approximation technique [108]. The most common models considered are linear and quadratic. Linear models require, for instance, $(n + 1)$ interpolation points but they disregard any

curvature information on the function. On the other hand, quadratic models require, in general, $(n+1)(n+2)/2$ interpolation points, which can be computationally expensive depending on the problem's dimension. In the papers [71] and [72], Powell constructs quadratic models using fewer points and shows empirically that it is possible to have efficient practical algorithms with $(2n+1)$ sample points. In [82], the authors address the importance of geometric conditions of the interpolation points to obtain global convergence of the algorithm. On the other hand, [34] claims to be possible to obtain a competitive algorithm even when omitting the geometry phase. There are many references in the literature that study DFO trust-region model-based algorithms. For unconstrained problems we can cite [17, 18, 19, 20, 34]. In particular, [12, 13] deal with partially separable objective functions, when the Hessian is sparse; [42] investigates the worst case function evaluations complexity for trust-region algorithms with linear interpolation models; [110, 111] rely on radial basis function interpolation models with a linear polynomial tail; [36] proposes a globally convergent algorithm, using the ideas from [70], that avoids unnecessary reductions of the trust-region radius. For box constrained problems, [71] considers linear and quadratic interpolation models and [43] uses recursive model-based active-set trust-region methods. In this context, [80] presents a review of derivative-free algorithms followed by a numerical comparison of 22 implementations using a test set of 502 problems, including convex, nonconvex, smooth and nonsmooth bounded problems. The references [60, 73] propose an algorithm for solving linear constrained problems, [15] presents a general approach for convex constrained problems, [85] considers problems in a convex, closed and bounded subset of a real Hilbert space. For solving general constrained optimization problems, [68] proposes a trust-region interpolation based-model algorithm with linear approximations to the constraint functions and [35] presents an algorithm that mixes an inexact restoration framework with filter techniques, where the optimality step is computed by trust-region algorithms.

Our approach belongs to the category of model-based methods. As the objective function in (1.2) is a probability, it has a particular structure: it is a componentwise nondecreasing function whose image is the closed unit interval. These properties should be

exploited by modelling the probability function by functions that share the same structure and are easier to evaluate, which motivated us to employ models based on copulæ. A copula is a multivariate probability distribution for which the marginal-probability distribution of each variable is uniform [29, 63]. The key result that connects copulæ to the probability function φ is the Sklar's theorem, which states that there exists a copula such that the probability function can be written as the composition of this copula with the univariate marginal distributions of the probability function. By this result, the multivariate probability function can be splitted into two independent parts: one describing the univariate marginal behaviour and the other, the dependence structures among the random variables [84]. As the marginal distributions are given or easy to estimate, our main task is to investigate these dependence structures through copulæ.

Modelling high-dimensional distribution functions is a challenging issue in many applications because it is not trivial to capture the dependence among the random variables. In optimization problems involving a probability function, copulæ were used in [25] as an alternative to the hard-to-evaluate function φ, and in [97] to model distributionally robust optimization problems. The direction we pursue in this work differs from [25]: instead of replacing φ by a single copula, which is a problem-dependent approach and involves a non-trivial statistical work of estimation, we consider a set/dictionary of copulæ to define a model that fits φ. Essentially, the proposed derivative-free trust-region method updates, at each iteration, a copula-based model by solving a least-square quadratic program. This iterative process of updating the model makes it capture by itself the dependence structures between the marginal distributions of the probability function, assigning weights to the copulæ in the dictionary and then building the model that best represents these dependencies.

Our DFO method for PMPs builds upon [15], but differs from the latter in the definition of the model and iterates. While [15] computes iterates as stationary points of quadratic constrained programs, our approach defines iterates as (approximate) stationary points of nonlinear optimization problems, i.e., the maximization of the copula-based model over a trust-region intersecting X. This shortcoming is compensated by the fact that our approach

uses an easy-to-evaluate model that approximates well the objective over the trust-region and, thus, relatively few (expensive) function evaluations are expected to be performed. This is indeed evidenced by the numerical experiments reported in Chapter 4, where the numerical performance of the new approach is compared to other DFO algorithms on several instances of academic and real-life probability maximization problems. In addition, based on a variant of our approach, we present preliminary numerical experiments of a heuristic for solving MINLP - Mixed Integer NonLinear Programming problems. In these experiments, the results are compared to the ones obtained by two other MINLP specialized solvers.

Contributions

The main contributions of this thesis are listed below. They have appeared in the article [10], recently published in the European Journal of Operational Research.

- The proposal of a DFO trust-region algorithm with copula-based models for solving Probability Maximization Problems.
- The presentation of a global convergence analysis of the proposed algorithm assuming reasonably mild hypotheses, most of which found in the DFO literature.
- Proposal of different strategies for constructing the copula-based models which led us to develop two implementable versions of the proposed algorithm.
- The presentation of extensive numerical experiments comparing the performance of our proposal with several DFO algorithms for solving academic and industrial probabilistic maximization problems.

Organization

The remainder of this work is organized as follows. In Chapter 2 we recall some basic concepts and main properties of probability theory and copulæ. Chapter 3 presents our derivative-free trust-region algorithm with models based on copulæ and analyzes its global

convergence. In Chapter 4, numerical experiments are reported. Chapter 5 concludes the manuscript with final remarks and comments on future steps.

Chapter 2

Probability distribution functions

This chapter recalls some definitions, results, and properties of probability functions and copulæ. We restrict our presentation to the relevant topics for the following chapters, and omit mathematical proofs for brevity. The interested reader is referred to the following articles [33, 77, 78, 81, 91, 92, 93, 99] and textbooks [3, 29, 30] for further discussions and mathematical proofs.

2.1 Basic concepts

In this section we present some notations and basic definitions of probability space and distribution functions, which are necessary to introduce copulæ. First we focus in the one-dimensional space and then we generalize some results to finite higher dimensions.

Definition 2.1. *Given the set Ω, a σ-algebra $\mathcal{F}$ of Ω is a nonempty collection of subsets of Ω that satisfy:*

(i) $\emptyset$ *and* Ω *belong to* $\mathcal{F}$*;*

(ii) if $A \in \mathcal{F}$*, then* $A^c \in \mathcal{F}$*;*

(iii) if $A_i \in \mathcal{F}$*,* $i \in \mathbb{N}$*, is a countable sequence of sets, then* $\bigcup_{i \in \mathbb{N}} A_i \in \mathcal{F}$*.*

Some examples of σ-algebras are the following [3]:

(a) Let Ω be any set and $\mathcal{F}$ be the family of all subsets of Ω;

(b) Let $\Omega = \mathbb{N}$ and let $\mathcal{F} = \big\{\emptyset,\ \{1,3,5,\ldots\},\ \{2,4,6,\ldots\},\ \Omega\big\}$;

(c) Let $\mathcal{F}$ be the family consisting of only two subsets of Ω, namely $\emptyset$ and Ω.

(d) Let $\Omega = \mathbb{R}$. The **Borel algebra** is the σ-algebra $\mathcal{B}(\mathbb{R})$ generated by all open intervals $]a, b[$ in $\mathbb{R}$. Observe that it is also the σ-algebra generated by all closed intervals $[a, b]$ in $\mathbb{R}$, by item (ii) of Definition 2.1. Any set in $\mathcal{B}(\mathbb{R})$ is called a **Borel set**.

The pair $(\Omega, \mathcal{F})$ consisting of a set Ω and a σ-algebra $\mathcal{F}$ of Ω is called a *measurable space*, i.e., it is a space on which we can define a measure.

Definition 2.2. *A measure is an extended real-valued function μ defined on a σ-algebra $\mathcal{F}$ of Ω, that is, a function $\mu : \mathcal{F} \to \mathbb{R}$ such that*

(i) $\mu(A) \geq \mu(\emptyset) = 0$ *for all* $A \in \mathcal{F}$*;*

(ii) if $A_i \in \mathcal{F}$ *is a countable sequence of disjoint sets, then*

$$\mu(\cup_i A_i) = \sum_i \mu(A_i).$$

The condition *(ii)* in Definition 2.2 is called *countable additivity*. If $\mu(\Omega) = 1$ we call μ a *probability measure* and we denote it by $\mathbb{P}$, which is defined as $\mathbb{P} : \mathcal{F} \to [0, 1]$. Now we can define a probability space.

Definition 2.3. *Let $\mathcal{F}$ be a σ-algebra of Ω and $\mathbb{P}$ be a probability measure on $\mathcal{F}$, then the triple $(\Omega, \mathcal{F}, \mathbb{P})$ is called a probability space.*

The elements of a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ have the following meaning: Ω is a set of "outcomes", $\mathcal{F}$ is a set of "events" and $\mathbb{P} : \mathcal{F} \to [0, 1]$ is a function that assigns probability to events.

Example 2.4. *Consider an experiment that a fair coin is flipped once, then the possible outcome is either heads* $\{H\}$ *or tails* $\{T\}$*, i.e.,* $\Omega = \{H, T\}$ *is our sample space. The* σ*-algebra contains* $2^{|\Omega|} = 2^2 = 4$ *elements/events, i.e.,* $\mathcal{F} = \big\{\emptyset, \{H\}, \{T\}, \{H, T\}\big\}$*. It is known that there is a fifty percent chance of tossing heads or tails, so the probability measure of the events are* $\mathbb{P}[\emptyset] = 0$*,* $\mathbb{P}[\{H\}] = 0.5$*,* $\mathbb{P}[\{T\}] = 0.5$ *and* $\mathbb{P}[\{H, T\}] = 0$*.*

An important concept in the probability theory that is necessary to define distribution functions is the *random variable*.

Definition 2.5. *A real valued function* $\boldsymbol{\xi}$ *defined on* Ω *is said to be a random variable if for every Borel set* $B \subset \mathbb{R}$ *we have* $\boldsymbol{\xi}^{-1}(B) = \{\omega \in \Omega : \boldsymbol{\xi}(\omega) \in B\} \in \mathcal{F}$*.*

Definition 2.6. *Consider the random variables* $\xi_1, \ldots, \xi_m$ *defined on the same probability space* $(\Omega, \mathcal{F}, \mathbb{P})$*. An m-dimensional random vector* $\xi = \{\xi_1, \ldots, \xi_m\}$ *is a measurable mapping from* Ω *into* $\mathbb{R}^m$*. In this case, the word "measurable" means that the counterimage*

$$\xi^{-1}(B) := \{\omega \in \Omega : \xi(\omega) \in B\}$$

of every Borel set B *in* $\mathcal{B}(\mathbb{R}^m)$ *belongs to* $\mathcal{F}$*.*

It can be proved that a random vector can be represented in the form $\xi = (\xi_1, \ldots, \xi_m)$, where $\xi_1, \ldots, \xi_m$ are one-dimensional random variables.

Given $\boldsymbol{\xi}$ a random variable on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, a probability measure $\mathbb{P}_{\boldsymbol{\xi}}$ may be defined on the measurable space $\big(\mathbb{R}, \mathcal{B}(\mathbb{R})\big)$ by

$$\mathbb{P}_{\boldsymbol{\xi}}[B] := \mathbb{P}\left[\boldsymbol{\xi}^{-1}(B)\right], \quad \forall B \in \mathcal{B}(\mathbb{R}).$$

The probability measure $\mathbb{P}_{\boldsymbol{\xi}}$ is called the *law*, *distribution* of $\boldsymbol{\xi}$, the *image probability of* $\mathbb{P}$ *under* $\boldsymbol{\xi}$ or the *cumulative distribution function of* $\boldsymbol{\xi}$. A similar construction applies to a random vector ξ, the only difference, in the vector case, being that the image probability is defined on $\big(\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m)\big)$.

Definition 2.7. *[29, Def. 1.2.9] The distribution function F_ξ of a random vector $\xi = (\xi_1, \ldots, \xi_m)$ on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is defined by, for all $x = (x_1, \ldots, x_m) \in \mathbb{R}^m$,*

$$F_\xi(x_1, \ldots, x_m) := \mathbb{P}\left[\xi_1 \leq x_1, \ldots, \xi_m \leq x_m\right].$$

In the literature it is common to use the function F_ξ or the notation $\xi \sim F$ to say that the functions F_ξ or F represent the distribution function of the random vector ξ. The next theorem shows that a distribution function can be characterized in terms of its analytical properties. In the sequence, $\mathbb{I}$ denotes the unit interval, i.e., $\mathbb{I} := [0, 1]$.

Theorem 2.8. *[29, Thm. 1.2.13] Let $F : \mathbb{R}^m \to \mathbb{I}$. The following statements are equivalent:*

- *there exists a random vector ξ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ such that F is the distribution function of ξ;*
- *F satisfies the following properties:*
  - *(a) For every $j \in \{1, \ldots, m\}$ and for all $x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_m$ in $\mathbb{R}$, the function*

$$t \mapsto F(x_1, \ldots, x_{j-1}, t, x_{j+1}, \ldots, x_m)$$

  is right-continuous;
  - *(b) F is m-increasing;*
  - *(c) $F(x) \to 0$, if at least one of the arguments of x tends to $-\infty$;*
  - *(d) $\lim\limits_{min\{x_1,\ldots,x_m\}\to+\infty} F(x_1, \ldots, x_m) = 1.$*

From item (b) of Theorem 2.8 we have the following result: if F is a distribution function, then for every $j \in \{1, \ldots, m\}$ and for all $x_1, \ldots, x_m$ in $\mathbb{R}$, the functions

$$t \in \mathbb{R} \mapsto F(x_1, \ldots, x_{j-1}, t, x_{j+1}, \ldots, x_m)$$

are increasing.

A definition that will be important throughout this text is the *marginal distribution* of a given distribution function F.

Definition 2.9. *[29, Def. 1.2.15] Let F be a m-dimensional distribution function of the random vector ξ and $\vartheta = (j_1, \ldots, j_d)$ a subvector of $(1, \ldots, m)$, $1 \leq d \leq m-1$. We call $\vartheta-$marginal of F the distribution function $F^{\vartheta} : \mathbb{R}^d \to \mathbb{I}$ defined by setting $m-d$ arguments of F equal to $+\infty$, namely, for all $u_1, \ldots, u_d$ in $\mathbb{I}$,*

$$F^{\vartheta}(u_1, \ldots, u_d) = F(v_1, \ldots, v_m),$$

where $v_j = u_j$ if $j \in \{j_1, \ldots, j_d\}$, and one lets v_j tend to $+\infty$ otherwise.

As is known, the marginal F^{ϑ} of the random vector $\xi \sim F$ is the joint distribution function of $(\xi_{j_1}, \ldots, \xi_{j_d})$. A particular case of interest is when $d = 1$, where the j-th 1-marginal of F_ξ, the distribution of ξ, is the 1-dimensional distribution function $F_{\xi_j} : \mathbb{R} \to \mathbb{I}$ of ξ_j and can be represented by

$$\begin{aligned} F_{\xi_j}(x_j) &= \lim_{(x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_m)\to(+\infty,\ldots,+\infty)} F_\xi(x_1, \ldots, x_m) \\ &= F_\xi(+\infty, \ldots, +\infty, x_j, +\infty, \ldots, +\infty). \end{aligned}$$

If the random variables $\xi_1, \ldots, \xi_m$ are independent and if F_{ξ_j} denotes the distribution function of ξ_j, $j = 1, \ldots, m$, then the distribution function of the random vector $\xi = (\xi_1, \ldots, \xi_m)$ can be written as the product of the marginals

$$F_\xi(x_1, \ldots, x_m) = \prod_{j=1}^{m} F_{\xi_j}(x_j).$$

A random vector $\xi = (\xi_1, \ldots, \xi_m)$ is said to be *absolutely continuous* if there exists a positive and integrable function $f_\xi : \mathbb{R}^m \to \mathbb{R}_+$, called *density function*, such that

$$\int_{\mathbb{R}^m} f_\xi \, \mathrm{d}\lambda_m = 1,$$

where λ_m is the m-dimensional Lebesgue measure. Now, we formalize the definition of the distribution function F_ξ when the random vector ξ is absolutely continuous.

Definition 2.10. *Let $\xi = (\xi_1, \ldots, \xi_m)$ be a random vector and F_ξ its distribution function. If there exists a function $f_\xi : \mathbb{R}^m \to \mathbb{R}_+$ such that, for all $(x_1, \ldots, x_m) \in \mathbb{R}^m$,*

$$F_\xi(x_1, \ldots, x_m) = \int_{-\infty}^{x_m} \ldots \int_{-\infty}^{x_1} f_\xi(t_1, \ldots, t_m)\ dt_1 \ldots dt_m,$$

then f_ξ is called density of the random vector ξ or joint density of the random variables $\xi_1, \ldots, \xi_m$ and, in this case, we say that $(\xi_1, \ldots, \xi_m)$ is absolutely continuous.

Now we give some examples of densities and their respective distributions functions (when a closed-form expression exists) in one-dimensional case. We illustrate the graphs of both functions in Figures 2.1, 2.2 and 2.3 for the uniform, exponential and normal distributions, respectively.

Example 2.11 (Uniform distribution on $\mathbb{I}$)**.** *Let $f(x) = 1$, for $x \in [0, 1]$, and 0, otherwise, be the density function. The distribution function F is:*

$$F(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } 0 \leq x \leq 1 \\ 1, & \text{if } x > 1. \end{cases}$$

We use the notation $F \sim U\big([0, 1]\big)$ to say that F follows a uniform distribution on $[0, 1]$.

Example 2.12 (Exponential distribution with rate λ)**.** *Let $f(x) = \lambda e^{-\lambda x}$, for $x \geq 0$, and 0, otherwise, be the density function. The distribution function F is:*

$$F(x, \lambda) = \begin{cases} 0, & \text{if } x \leq 0 \\ 1 - e^{-\lambda x}, & \text{if } x \geq 0. \end{cases}$$

We use the notation $F \sim \exp(\lambda)$ to say that F follows an exponential distribution with parameter λ.

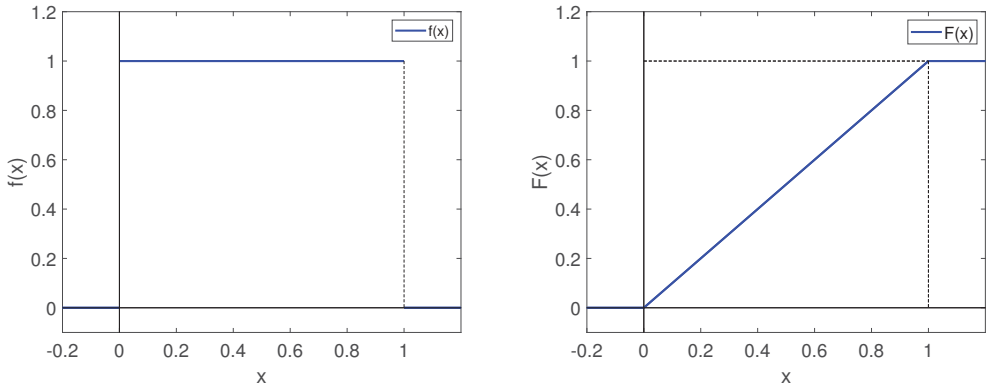


Figure 2.1: Uniform density (left) and distribution (right) functions on $\mathbb{I}$.

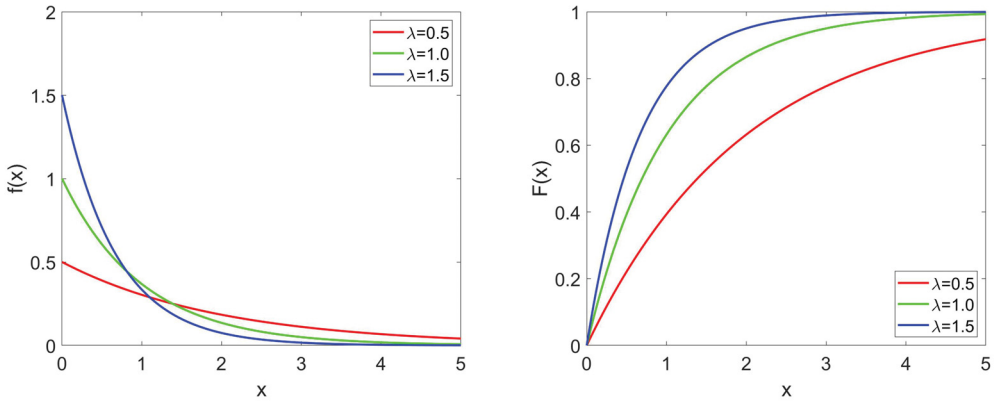


Figure 2.2: Exponential density (left) and distribution (right) functions with different parameters λ.

Example 2.13 (Standard normal distribution)**.** *Let* $f(x) = \dfrac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}$*, for* $x \in \mathbb{R}$*, be the density function.*

In this case, there is no analytic function for the distribution function $F(x)$*. We use the notation* $F \sim N(\mu, \sigma^2)$ *to say that* F *follows a normal distribution with mean or expectation* μ *and standard deviation* σ *(or variance* σ^2*).*

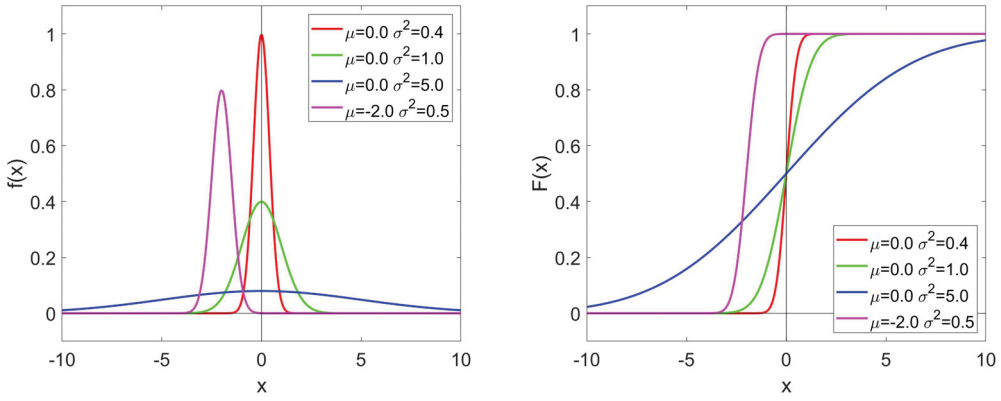


Figure 2.3: Normal density (left) and distribution (right) functions with different parameters μ and σ^2.

2.2 Continuity and differentiability of distribution functions

In this section we discuss and present classical results about some analytical properties of probability functions, such as continuity and differentiability. There are many references in the literature related to these subjects and some of them are [33, 77, 78, 81, 91, 92, 93, 99, 102].

2.2.1 Continuity

When analysing properties of a (probability) function, a first question that may arise is under which conditions it is continuous. In other words, the continuity of the distribution function of the random vector ξ. First of all, we define continuity properties of set-valued mappings.

Definition 2.14. *[81, Def. 5.4] Let $M : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ be a set-valued mapping. The mapping M is said to be outer semicontinuous at $\bar{x} \in \mathbb{R}^n$ if*

$$\limsup_{x \to \bar{x}} M(x) \subseteq M(\bar{x}),$$

or equivalently $\limsup_{x\to\bar{x}} M(x) = M(\bar{x})$*, which means that any (possible) cluster point* z *of* $\{z_n\}_{n\geq 0}$ *must belong to* $M(\bar{x})$*, where* $z_n \in M(x_n)$ *and* $x_n \to \bar{x}$*. The mapping* M *is said to be inner semicontinuous at* $\bar{x} \in \mathbb{R}^n$ *if*

$$M(\bar{x}) \subseteq \liminf_{x\to\bar{x}} M(x),$$

or equivalently when M *is closed-valued,* $\liminf_{x\to\bar{x}} M(x) = M(\bar{x})$*.* M *is called continuous at* $\bar{x}$ *if both conditions hold, i.e., if* $M(x) \to M(\bar{x})$*, as* $x \to \bar{x}$*.*

The interested reader can find more content about the equivalences of Definition 2.14, continuity and semicontinuity properties of set-valued mappings in the book [81].

To relate set-valued mappings to the probability function, we consider the equivalent formulation of the probability function by letting $M : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ and

$$\varphi(x) := \mathbb{P}\left[\xi \in M(x)\right]. \tag{2.1}$$

Now we present a continuity result in two forms, one related to set-valued mappings and the other makes explicit reference to a function $g : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^k$ defining the set-valued mapping $M(x)$.

Proposition 2.15. *[92, Prop. 2.1] Assume that the set-valued application* $M : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ *is both outer and inner semicontinuous and convex-valued. If moreover for an arbitrary* $x \in \mathbb{R}^n$

$$\mathbb{P}\left[\xi \in bd\ M(x)\right] = 0$$

where bd M *denotes the boundary of set* M *in* $\mathbb{R}^m$*, then* φ*, given in* (2.1)*, is continuous at any* $x \in \mathbb{R}^n$*. If the set-valued application* M *is only outer semicontinuous, then the probability function* φ *is upper semicontinuous.*

We say that M is *convex-valued* if $M(x)$ is a convex set for each $x \in \mathbb{R}^n$.

Proposition 2.16. *[92, Prop. 2.2] Let* $g : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^k$ *be a continuous mapping and*

assume that the following regularity condition holds for all $x \in \mathbb{R}^n, j = 1, \ldots, k$:

$$\mathbb{P}\left[g_j(x, \xi) = 0\right] = 0,$$

then the probability function $\varphi = \mathbb{P}\left[g(x, \xi) \leq 0\right]$ *is continuous at any* $x \in \mathbb{R}^n$. *If the mapping* g *is lower semicontinuous, then* φ *is upper semicontinuous.*

Note that these results hold for a probability function even more general than the one we are considering, where g depends only on the decision variable x. The condition of M being convex-valued and the regularity assumptions in Proposition 2.15, and in 2.16, for the function g, are not so restrictive.

An example illustrating the technical necessity of the given regularity condition is presented in [92, Example 2.1], where a simple reformulation fixed the discontinuity of the probability function and let it infinitely differentiable. Also, [92] summarizes how to ensure the regularity condition, by assuming both:

- ξ has a density with respect to Lebesgue measure;
- $\{z \in \mathbb{R}^m : g(x, z) = 0\}$ is a Lebesgue null set. This last condition is, for instance, satisfied if

$$bd\ \{z \in \mathbb{R}^m : g(x, z) \leq 0\} = \{z \in \mathbb{R}^m : g(x, z) = 0\}. \tag{2.2}$$

The regularity conditions on M and g in Propositions 2.15 and 2.16, respectively, are linked by these assumptions. Another way to ensure (2.2) is by considering that g is convex in the second argument (which is valid in our case) and admits a Slater point.

2.2.2 Differentiability

The differentiability property of the probability function is more restrictive than continuity, in which additional assumptions are needed. Even a simple example [99, Proposition 2.2], where $\varphi = \mathbb{P}\left[g(x, \xi) \leq 0\right]$ with nice input data (ξ following a regular Gaussian distribution, the function g is smooth and convex in the second argument and the inequality

defined by g satisfies the Slater condition) fails to be differentiable without the compactness of the set $M(x) := \{z \in \mathbb{R}^m : g(x,z) \leq 0\}$. Another example of a similar situation is given in [92, Example 2.3].

As discussed in [92], the differentiability of φ is investigated in two paths. The first one makes fairly few assumptions on the distribution of ξ, but more restrictive ones on everything else, given rise to relative general results. The second path focuses in particular distribution functions where suitable additional assumptions can be assumed. We first present a well-known result directed to the first path of investigation [89, 90].

Theorem 2.17. *[92, Theorem 2.1] Let $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k$ be a continuously differentiable function and let $\theta : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ be a continuously differentiable density. Pick an arbitrary $1 \leq l < k$. Assume moreover that*

1) The set $M(x) = \{z \in \mathbb{R}^m : g(x,z) \leq 0\}$ is bounded in a neighbourhood U of some point $\bar{x}$.

2) At $\bar{x}$ all constraints $g_i(\bar{x}, z) \leq 0$, $i = 1, \ldots, k$ are active, i.e., $M(\bar{x}) \cap \{z \in \mathbb{R}^m : g_i(\bar{x}, z) = 0\} \neq \emptyset$.

3) One can find a continuous matrix function $H_l : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{n \times m}$ satisfying

$$H_l(x,z)\nabla_z g^l(x,z) + \nabla_x g^l(x,z) = 0$$

where $g^l(x,z) = \big(g_1(x,z), \ldots, g_l(x,z)\big) \in \mathbb{R}^l$.

4) The matrix function H_l has a continuous partial derivative with respect to z.

5) The gradient $\nabla_z g_i(x,z) \neq 0$ on $\partial_i M(\bar{x}) := M(\bar{x}) \cap \{z \in \mathbb{R}^m : g_i(\bar{x}, z) = 0\}$.

6) For each $z \in M(\bar{x})$, the vectors $\nabla_z g_i(\bar{x}, z), i \in I(\bar{x}, z) := \{j : g_j(\bar{x}, z) = 0\}$ are linearly independent.

Then probability function $\varphi(x) := \int_{M(x)} \theta(x,z)d\lambda(z) = \mathbb{P}\left[\xi \in M(x)\right]$ *is differentiable at* $\bar{x}$ *and*

$$\begin{aligned}\nabla_x \varphi(\bar{x}) = &\int_{M(\bar{x})} \nabla_x \theta(\bar{x},z) + \operatorname{div}_z\left(\theta(\bar{x},z)H_l(\bar{x},z)\right)d\lambda(z) \\ &- \sum_{i=l+1}^{k} \int_{\partial_i M(\bar{x})} \frac{\theta(\bar{x},z)}{\|\nabla_z g_i(\bar{x},z)\|}\left[\nabla_x g_i(\bar{x},z) + H_l(\bar{x},z)\nabla_z g_i(\bar{x},z)\right] dS,\end{aligned} \tag{2.3}$$

where λ *is the Lebesgue measure on* $\mathbb{R}^m$.

The very technical proof of Theorem 2.17 is presented in [89]. A clearer idea of how to prove formula (2.3) is shown in the appendix of [90]. The compactness of the set $M(x)$ and the LICQ conditions can be replaced by an integrability and a pairwise independence request [92], respectively.

In the previous theorem one can choose the constant $1 \leq l < k$ in such a way that is more convenient for the application and two especial cases can also be explored, i.e., when $l = 0$ or $l = k$. In the first case the matrix H_l is absent and (2.3) is reduced to the integral over the volume, and when $l = k$ the formula is the integral over the surface.

Theorem 2.18. *[92, Theorem 2.2] Under the notation and conditions as in Theorem 2.17, let* $l = 0$. *Then, we have:*

$$\nabla_x \varphi(x) = \int_{M(x)} \nabla_x \theta(x,z) - \sum_{i=1}^{k} \int_{\partial_i M(x)} \frac{\theta(x,z)}{\|\nabla_z g_i(x,z)\|} \nabla_x g_i(x,z) dS.$$

If $l = k$ *we have:*

$$\nabla_x \varphi(x) = \int_{M(x)} \nabla_x \theta(x,z) + \operatorname{div}_z\left(\theta(x,z)H_k(x,z)\right)d\lambda(z).$$

In [88, 89, 91] some examples are presented demonstrating possible applications of the formulas from Theorems 2.17 and 2.18. These previous results are part of the first path and represent a state of the art if we do not assume any specific form of the random vector ξ, but their generality have a cost due to the difficulty in computing the matrix H_l or the

term $\operatorname{div}_z\big(\theta(x,z)H_l(x,z)\big)$.

In relation to the second path, we consider some special distributions of the random vector ξ, which allow us to obtain explicit results. First of all, when the probability function has the separable setting, as in (1.2), we can provide the following general result ensuring that each component of the partial derivative of F_ξ consists on computing a numerical integration of dimension $m-1$.

Theorem 2.19. *[77] Let $\xi \in \mathbb{R}^m$ be a random vector with density $f_\xi : \mathbb{R}^m \to \mathbb{R}$. Fix any $\bar{z} \in \mathbb{R}^m$ and consider $F_\xi(z) := \mathbb{P}[\xi \leq z]$. If*

$$\varphi^{(i)}(t) := \int_{-\infty}^{\bar{z}_1} \dots \int_{-\infty}^{\bar{z}_{i-1}} \int_{-\infty}^{\bar{z}_{i+1}} \dots \int_{-\infty}^{\bar{z}_m} f_\xi\left(u_1, \dots, u_{i-1}, t, u_{i+1}, \dots, u_m\right) du_1 \dots du_{i-1} du_{i+1} \dots du_m$$

is continuous for all $i = 1, \dots, m$, then $F_\xi(z)$ is partially differentiable at $\bar{z}$ and

$$\frac{\partial F_\xi}{\partial z_i}(\bar{z}) = \varphi^{(i)}\left(\bar{z}_i\right).$$

From Theorem 2.19 we can obtain the following differentiability result for the Gaussian distribution [47, 49, 77, 78].

Lemma 2.20. *[93, Lem. 2.7.5] Let ξ be an m-dimensional Gaussian random vector with mean $\mu \in \mathbb{R}^m$ and positive definite variance-covariance matrix Σ. Then the distribution function $F_\xi(z) := \mathbb{P}[\xi \leq z]$ is continuously differentiable and in any fixed $z \in \mathbb{R}^m$ the following holds:*

$$\frac{\partial F_\xi}{\partial z_i}(z) = f_{\xi_i}\left(z_i\right) F_{\tilde{\xi}(z_i)}\left(z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_m\right), \ i = 1, \dots, m. \tag{2.4}$$

Here $\tilde{\xi}\left(z_i\right)$ is a Gaussian random variable with mean $\hat{\mu} \in \mathbb{R}^{m-1}$ and $(m-1) \times (m-1)$ positive definite covariance matrix $\hat{\Sigma}$. Let D_m^i denote the m-th order identity matrix from which the ith row has been deleted. Then

$$\hat{\mu} = D_m^i\left(\mu + \Sigma_{ii}^{-1}\left(z_i - \mu_i\right)\Sigma_i\right) \text{ and } \hat{\Sigma} = D_m^i\left(\Sigma - \Sigma_{ii}^{-1}\Sigma_i\Sigma_i^\top\right)\left(D_m^i\right)^\top,$$

where Σ_i is the i-th column of Σ and Σ_{ii} is the ith element of the main diagonal of Σ.

Note that Lemma 2.20 requires a positive definite covariance matrix Σ, i.e., the Gaussian random vector ξ must be non-degenerate. This hypothesis restricts many applications, because in some of them occur a multiplication of a non-degenerate Gaussian random variable with a matrix that has more lines than columns, causing the degeneracy. The next two results are important generalizations of Lemma 2.20.

Lemma 2.21. *[93, Lem. 2.7.6] Let A be a $k \times m$ matrix. Consider a linear inequality system $Ax \leq z$ and define*

$$\mathcal{I}(A,z) = \left\{ I \subseteq \{1,\ldots,k\} : \exists x \in \mathbb{R}^m, a_i^\top x = z_i, i \in I, a_i^\top x < z_i, i \notin I \right\}$$

Assume that $z \in \mathbb{R}^m$ is such that $Ax \leq z$ is non-degenerate (i.e., rank $\{a_i\}_{i\in I} = |I|\ \forall I \in \mathcal{I}(A,z)$). Let ξ be an m-dimensional Gaussian random vector with mean μ and positive definite variance-covariance matrix Σ. Then the probability function $\varphi(z) = \mathbb{P}[A\xi \leq z]$ is differentiable at z and

$$\frac{\partial \varphi}{\partial z_j}(z) = \begin{cases} 0 & \text{if} \quad \{j\} \notin \mathcal{I}(A,z) \\ f_j(z_j)\, \mathbb{P}\left[A^{(j)} L^{(j)} \xi^{(j)} \leq z^{(j)} - A^{(j)} w^{(j)}\right] & \text{if} \quad \{j\} \in \mathcal{I}(A,z) \end{cases}.$$

Here $\xi^{(j)}$ is a centered $m-1$ dimensional Gaussian random variable with independent components, $A^{(j)}$ is obtained from A by deleting row j, $z^{(j)}$ is defined similarly. Moreover, $L^{(j)}$ is the Choleski matrix of $S^{(j)} := \Sigma - \frac{1}{a_j^\top \Sigma a_j} \Sigma a_j a_j^\top \Sigma$ (i.e., $S^{(j)} = L^{(j)} \left(L^{(j)}\right)^\top$), $w^{(j)} = \mu + \frac{z_j - a_j^\top \mu}{a_j^\top \Sigma a_j} \Sigma a_j$ and f_j the one-dimensional Gaussian density with mean $\mu^\top a_j$ and variance $a_j^\top \Sigma a_j$. Finally the inequality system $A^{(j)} L^{(j)} y \leq z^{(j)} - A^{(j)} w^{(j)}$ is non-degenerate.

An interesting observation about Theorem 2.21 is that if the original inequality system $Ax \leq z$ happens to be non-degenerate, and consequently the reduced one also is, then the reduced inequality system fulfills the assumptions of the same lemma, which allows one to obtain derivative formulas of any order recursively. In other words, considering that z satisfies the non degeneracy assumption, the probability function is of class $\mathcal{C}^\infty$ [49]. A

similar idea, weakening the positive definiteness of the covariance matrix, can be applied to singular Gaussian distributions at any points z satisfying the non degeneracy condition.

Theorem 2.22. *[49, Theorem 4.1] Let $\xi \sim \mathcal{N}(\mu, \Sigma)$ with some (possibly singular) covariance matrix $\Sigma = (\sigma_{ij})$ of order (m, m). Denote by $\Sigma = AA^T$ any factorization of the positive semidefinite matrix Σ. Let z be such that the inequality system $Ax \leq z - \mu$ is non-degenerate. Then, for $j = 1, \ldots, m$ one has the formula*

$$\frac{\partial F_\xi}{\partial z_j}(z) = f_{\xi_j}(z_j) \cdot F_{\tilde{\xi}(z_j)}(z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_m).$$

Here, f_{ξ_j} denotes the one-dimensional Gaussian density of the component ξ_j, $\tilde{\xi}(z_j)$ is an $(m-1)$-dimensional (possibly singular) Gaussian random vector distributed according to $\tilde{\xi}(z_j) \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma})$, $\hat{\mu}$ results from the vector $\mu + \sigma_{jj}^{-1}(z_j - \mu_j)\sigma_j$ by deleting component j, and $\hat{\Sigma}$ results from the matrix $\Sigma - \sigma_{jj}^{-1}\sigma_j\sigma_j^T$ by deleting row j and column j, where σ_j refers to column j of Σ.

A remarkable case with a special structure of (1.2) commonly arises in energy management problems, which is one of the examples considered in the numerical experiments in Chapter 4. Such structure is composed of a bilateral inequality within the probability function and is given by

$$\varphi(x) := \mathbb{P}[Ax + a \leq \xi \leq Bx + b], \tag{2.5}$$

where $\xi \in \mathbb{R}^m$ is a random vector and the vectors $a, b \in \mathbb{R}^m$ and matrices $A, B \in \mathbb{R}^{m \times n}$ are deterministic. The inequality system can be reformulated to a unilateral one as follows:

$$\begin{pmatrix} I \\ -I \end{pmatrix} \xi \leq \begin{pmatrix} B \\ -A \end{pmatrix} x + \begin{pmatrix} b \\ -a \end{pmatrix}, \tag{2.6}$$

where $I \in \mathbb{R}^{m \times m}$ is the identity matrix. The disadvantage of this reformulation is that the

new random vector

$$\bar{\xi} := \begin{pmatrix} I \\ -I \end{pmatrix} \xi \in \mathbb{R}^{2m} \tag{2.7}$$

is degenerate and Lemma 2.20 can not be applied. However, Theorem 2.21 fits in this case and provides a differentiability formula for the distribution of $\bar{\xi}$. The price for doubling the dimension of the random vector is paid by evaluating the probability in dimension $2m$, which is much more expensive. The following result from [106] has the advantage of not working with probability in such dimension.

Theorem 2.23. *[106, Thm. 1] Assume that $\xi \sim \mathcal{N}(\mu, \Sigma)$ with some positive definite covariance matrix Σ. Then, for $i = 1, \ldots, m$,*

$$\frac{\partial}{\partial b_i} F_\xi(a,b) = f_{\xi_i}(b_i)\, F_{\tilde{\xi}(b_i)}(\tilde{a}, \tilde{b})$$
$$\frac{\partial}{\partial a_i} F_\xi(a,b) = -f_{\xi_i}(a_i)\, F_{\tilde{\xi}(a_i)}(\tilde{a}, \tilde{b}).$$

Here, f_{ξ_i} is as in Lemma 2.20, $\tilde{\xi}(b_i), \tilde{\xi}(a_i)$, are $m-1$-dimensional random vectors distributed according to $\tilde{\xi}(b_i), \tilde{\xi}(a_i) \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma})$ such that $\hat{\mu}$ results from the vector $\mu + \sigma_{ii}^{-1}(b_i - \mu_i)\sigma_i$ (in case of b_i) or from the vector $\mu + \sigma_{ii}^{-1}(a_i - \mu_i)\sigma_i$ (in case of a_i) by deleting component i and $\hat{\Sigma}$ is defined as in Lemma 2.20. Moreover $\tilde{a}$ and $\tilde{b}$ result from a and b by deleting the respective component i.

A formula for the derivative of the probability function in (2.5) is obtained by combining the previous Lemma with the Corollary 2.24 that follows.

Corollary 2.24. *[93, Cor. 3.2.3] Let $\varphi : \mathbb{R}^n \to [0,1]$ be defined as $\varphi(x) := \mathbb{P}[Ax + a \leq \xi \leq Bx + b]$, where $\xi \in \mathbb{R}^m$ is a Gaussian random variable with mean $\mu \in \mathbb{R}^m$ and positive definite variance-covariance matrix Σ. Moreover, let a, b, A, B be as in (2.5) Then the mapping φ is twice differentiable and we have:*

$$\nabla\varphi = \nabla_a F_\xi(a,b)^\top A + \nabla_b F_\xi(a,b)^\top B$$
$$\nabla^2\varphi = A^\top \nabla^2_{aa} F_\xi(a,b) A + A^\top \nabla^2_{ab} F_\xi(a,b) B + B^\top \nabla^2_{ba} F_\xi(a,b) A + B^\top \nabla^2_{bb} F_\xi(a,b) B$$

where F_ξ is defined as in Lemma 2.23.

Two other special cases allowing the computation of gradients are the multivariate Gamma [77, 75] and Dirichlet [41, 77, 113] distributions, given by Theorem 2.25 and Theorem 2.26, respectively.

Theorem 2.25. *[93, Thm. 2.7.7] A multivariate Gamma distribution $\zeta \in \mathbb{R}^m$ is defined as $\zeta = A\eta$, where $\eta \in \mathbb{R}^{2^m-1}$ contains independent standard Gamma (with parameters ϑ_j) distributed components and A is a $m \times 2^m - 1$ matrix with non-zero columns, $A_{ij} \in \{0,1\}$ for $i = 1,\ldots,m,\ j = 1,\ldots,2^m-1$. Define, for each $i = 1,\ldots,m,\ I_i \subseteq \{1,\ldots,2^m-1\}$ as $I_i = \{j : A_{ij} = 1\}$. Then $\delta^i \in \mathbb{R}^{m-1}$, where*

$$\delta_k^i = \frac{\sum_{j\in I_k\cap I_i}\eta_j}{\sum_{j\in I_i}\eta_j},\ k = 1,\ldots,i-1,i+1,\ldots,m,$$

is an $m-1$ dimensional Dirichlet Distribution with parameters

$$\Theta_k = \sum_{j\in I_k\cap I_i}\vartheta_j,\ k = 1,\ldots,i-1,i+1,\ldots,m$$
$$\Theta_{m+1} = \sum_{j\in\cup_{k\neq i}I_k\setminus I_i}\vartheta_j,$$

for each $i = 1,\ldots,m$. Now $F(z) := \mathbb{P}[\zeta \le z]$ is partially differentiable and

$$\frac{\partial F_\xi}{\partial z_i}(z) = \mathbb{P}\left[z_i\delta_k^i + \gamma_k \le z_k\ \forall k \neq i\right]\frac{z_i^{\vartheta_i-1}e^{-z_i}}{\Gamma(\vartheta_i)}$$

where $\gamma_k = \sum_{j\in I_k\cap\bar{I}_i}\eta_j,\ k = 1,\ldots,i-1,i+1,\ldots,m$, is an $m-1$ dimensional multivariate gamma distribution independent of δ^i, $\bar{I}_i$ is the complement of I_i and Γ is the usual gamma-function.

Theorem 2.26. *[93, Thm. 2.7.8] Let $\xi \in \mathbb{R}^m$ have a multivariate Dirichlet distribution, i.e., have the density:*

$$f(z_1,\ldots,z_m) = \frac{\Gamma(\vartheta_1+\ldots+\vartheta_{m+1})}{\Gamma(\vartheta_1)\cdots\Gamma(\vartheta_{m+1})}z_1^{\vartheta_1-1}\cdots z_m^{\vartheta_m-1}\left(1-\sum_{j=1}^m z_j\right)^{\vartheta_{m+1}-1},$$

on the unit simplex $z \in \Delta_m = \{z \in \mathbb{R}^m \mid z_1 + \cdots + z_m = 1 \text{ and } z_i \geq 0, i = 1, \ldots, m\}$ *in dimension* m *(zero elsewhere). If*

$$y^i = \left(\frac{z_1}{1 - z_i}, \ldots, \frac{z_{i-1}}{1 - z_i}, \frac{z_{i+1}}{1 - z_i}, \ldots, \frac{z_m}{1 - z_i} \right) \in \mathbb{R}^{m-1}$$

satisfies $y^{(1)} + y^{(2)} > 1$ *or* $y^{(1)} + y^{(2)} + y^{(3)} > 1$ *(but* $y^{(1)} + y^{(2)} \leq 1$*) for the order-statistics* $y^{(\cdot)}$*, then* $F(z) := \mathbb{P}[\xi \leq z]$ *is partially differentiable at* z *and*

$$\frac{\partial F_\xi}{\partial z_i}(z) = \mathbb{P}\left[\tilde{\xi}^i_k \leq y^i_k, \forall k \neq i \right] \frac{\Gamma(\vartheta_1 + \ldots + \vartheta_{m+1})}{\Gamma(\vartheta_i)\, \Gamma\left(\sum_{j \neq i} \vartheta_j \right)} z_i^{\vartheta_i - 1} (1 - z_i)^{\sum_{j \neq i} \vartheta_j - 1}$$

where $\tilde{\xi}^i$ *has an* $m - 1$ *dimensional Dirichlet distribution with parameters* $\vartheta_1, \ldots, \vartheta_{i-1}$ $\vartheta_{i+1}, \ldots, \vartheta_{m+1}$.

A promising family that deserves our attention are the elliptically symmetric distributions, where some examples are the multivariate Gaussian, Student, logistic or exponential power random vectors [53]. We refer the book [33] and [11, 53, 66, 100, 101] for the interested reader in the subject.

Definition 2.27. *[94, Def. 1] We say that the random vector* $\xi \in \mathbb{R}^m$ *is elliptically symmetrically distributed with mean* μ*, covariance matrix* Σ *and generator* $\theta : \mathbb{R}_+ \to \mathbb{R}_+$*, notation* $\xi \sim \mathcal{E}(\mu, \Sigma, \theta)$ *if and only if its density* $f_\xi : \mathbb{R}^n \to \mathbb{R}_+$*is given by:*

$$f_\xi(z) = \theta\Big((z - \mu)^T \Sigma^{-1} (z - \mu) \Big) / \sqrt{\det \Sigma} \tag{2.8}$$

Two examples of generators associated to the respective distribution function are given in the following.

Example 2.28. *[102] The Gaussian and Student random vectors are elliptical with the respective generators:*

$$\theta^{Gauss}(t) = \exp(-t/2)/(2\pi)^{m/2},$$

$$\theta^{Student}(t) = \frac{\Gamma\left(\frac{m+\nu}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} (\pi\nu)^{-m/2} \left(1 + \frac{t}{\nu}\right)^{-\frac{m+\nu}{2}}.$$

In Definition 2.27, we emphasize that our attention is restricted to random vectors disposing of a density, which is appropriate for our configuration, but not necessarily of full generality. The characteristic function provides an alternative way for describing a random vector, determining the behavior and properties of its probability distribution. For $\xi \sim \mathcal{E}(\mu, \Sigma, \theta)$, the characteristic function is defined by the first equality of (2.9) and it can also be represented by the second equality of (2.9)

$$\varphi_\xi(t) = \mathbb{E}\left(\exp\left(it^\top \xi\right)\right) = \exp\left(it^\top \mu\right) \psi\left(t^\top \Sigma t\right) \tag{2.9}$$

for a scalar mapping ψ, called characteristic generator, which is defined as

$$\psi(v) = \int_0^\infty \mathbb{E}\left[\exp\left(i\sqrt{v} r \zeta_1\right)\right] 2\frac{\pi^{\frac{m}{2}}}{\Gamma\left(\frac{m}{2}\right)} r^{m-1} \theta\left(r^2\right) dr,$$

where $\zeta \in \mathbb{R}^m$ has uniform distribution on the Euclidean sphere $\mathbb{S}^{m-1} = \{z \in \mathbb{R}^m : \|z\|^2 = 1\}$ and ζ_1 denotes its first component. The expression (2.9) follows directly from the definition of a characteristic function considering a change of variables and, as a consequence of [33, Theorem 2.1], $L^{-1}(\xi - \mu)$ follows a spherical distribution, where L denotes the matrix arising from the Choleski decomposition of $\Sigma = LL^T$. From [33, Corollary to Theorem 2.2], $L^{-1}(\xi - \mu)$ admits the representation

$$L^{-1}(\xi - \mu) = \mathcal{R}\zeta, \tag{2.10}$$

where $\mathcal{R}$ is a one-dimensional random variable with support on $\mathbb{R}_+$ (corresponds to the smallest closed subset of $\mathbb{R}_+$ such that its probability distribution, according to $\mathcal{R}$, is 1), independent of ζ. Now, from (2.10) it follows that ξ admits the representation

$$\xi = \mu + \mathcal{R}L\zeta. \tag{2.11}$$

Without loss of generality, we will assume that $\mu = 0$ and Σ is a correlation matrix. Indeed, define the random variable $\hat{\xi} := D(\xi - \mu)$, where D is an $m \times m$ diagonal matrix with

elements $D_{ii} = \Sigma_{ii}^{-1/2}$. We may have that $\hat{\xi} \sim \mathcal{E}(0, R, \theta)$, where R is the correlation matrix associated with Σ. By defining the mapping $\hat{g} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ as $\hat{g}(x, z) := g(x, D^{-1}z + \mu)$, where $\hat{g}$ has the same properties of g, the following identity holds

$$\begin{aligned} \varphi(x) &= \mathbb{P}\left[\hat{g}(x, \hat{\xi}) \leq 0\right] = \mathbb{P}\left[\hat{g}\big(x, D(\xi - \mu)\big) \leq 0\right] \\ &= \mathbb{P}\left[g(x, D^{-1}D(\xi - \mu) + \mu) \leq 0\right] = \mathbb{P}\left[g(x, \xi) \leq 0\right]. \end{aligned}$$

The advantage of representation (2.11) (with $\mu = 0$) is that for a given Lebesgue measurable set $M \subseteq \mathbb{R}^m$ its probability may be represented as

$$\mathbb{P}\left[\xi \in M\right] = \int\limits_{v \in \mathbb{S}^{m-1}} \mu_{\mathcal{R}}(\{r \geq 0 : rLv \cap M \neq \emptyset\}) d\mu_{\zeta}, \tag{2.12}$$

where $\mu_{\mathcal{R}}$ and μ_{ζ} are the measures associated with $\mathcal{R}$ and ζ, respectively. The set M can be assumed as the set-valued application $M(x)$ of Proposition 2.15 or the inequality system g, as in Proposition 2.16.

Assuming the maximum function $g^{\mathrm{m}} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ over its components as

$$g^{\mathrm{m}}(x, z) = \max_{j=1,\ldots,k} g_j(x, z),$$

which preserves the convexity in the second argument (but not differentiability), the general probability function $\varphi = \mathbb{P}\left[g(x, \xi) \leq 0\right]$ can be written, by (2.12), as

$$\varphi(x) = \int\limits_{v \in \mathbb{S}^{m-1}} \mu_{\mathcal{R}}(\{r \geq 0 : g^{\mathrm{m}}(x, rLv) \leq 0\}) d\mu_{\zeta} = \int_{v \in \mathbb{S}^{m-1}} e(x, v) d\mu_{\zeta}, \tag{2.13}$$

where

$$e(x, v) = \mu_{\mathcal{R}}(\{r \geq 0 : g^{\mathrm{m}}(x, rLv) \leq 0\}),\ \forall x \in \mathbb{R}^n, \forall v \in \mathbb{S}^{m-1}. \tag{2.14}$$

In the following we will consider points x for which $g^{\mathrm{m}}(x, z) < 0$, that means 0 is a Slater point of the inequality system $g(x, z) \leq 0$ in z. This assumption with the convexity

of g^{m} imply that for each $x \in \mathbb{R}^n$ and each $v \in \mathbb{S}^{m-1}$, (2.14) can be simplified as

$$e(x, v) = \mu_{\mathcal{R}}([0, r^*]),$$

where $r^* = \infty$ in the case that $g^{\mathrm{m}}(x, rLv) < 0$ for all $r > 0$ or r^* is the unique solution of $g^{\mathrm{m}}(x, rLv) = 0$ in $r > 0$. Since these two cases are essential when dealing with possibly unbounded sets, we define the following set-valued mappings $F_j, I_j, F, I : \mathbb{R}^n \rightrightarrows \mathbb{S}^{m-1}$, for $j = 1, \ldots, k$:

$$\begin{aligned} F(x) &:= \left\{ v \in \mathbb{S}^{m-1} \mid \exists r > 0 : g^{\mathrm{m}}(x, rLv) = 0 \right\} \\ I(x) &:= \left\{ v \in \mathbb{S}^{m-1} \mid \forall r > 0 : g^{\mathrm{m}}(x, rLv) < 0 \right\} \\ F_j(x) &:= \left\{ v \in \mathbb{S}^{m-1} \mid \exists r > 0 : g_j(x, rLv) = 0 \right\} \\ I_j(x) &:= \left\{ v \in \mathbb{S}^{m-1} \mid \forall r > 0 : g_j(x, rLv) < 0 \right\}. \end{aligned}$$

We now address some elementary properties and then the differentiability results of φ by following the ideas presented in [93, 99], for the Gaussian case, and [94, Sections 2.4 and 3] for general distributions, which references to [99, 100].

Lemma 2.29. *[100, Lem 2.1] Let $x \in \mathbb{R}^n$ be such that $g^{\mathrm{m}}(x, 0) < 0$. Then,*

1. *$F_j(x) \cup I_j(x) = F(x) \cup I(x) = \mathbb{S}^{m-1}$ for all $j = 1, \ldots, k$.*

2. *For $j \in \{1, \ldots, k\}$ and $v \in F_j(x)$ let $r > 0$ be such that $g_j(x, rLv) = 0$. Then,*

$$\langle \nabla_z g_j(x, rLv), Lv \rangle \geq -\frac{g_j(x, 0)}{r}.$$

3. *$F(x) = \cup_{j=1}^k F_j(x), I(x) = \cap_{j=1}^k I_j(x)$.*

4. *$e(x, v) = 1$ if $v \in I(x)$ and $e(x, v) < 1$ if $v \in F(x)$.*

Lemma 2.30. *[99, Lem. 3.2] Let $j = 1, \ldots, k$ be arbitrary and let (x, v) be such that $g_j(x, 0) < 0$ and $v \in F_j(x)$. Then, there exist neighbourhoods U_j of x and V_j of v as well as a continuously differentiable function $\rho_j^{x,v} : U_j \times V_j \to \mathbb{R}_+$ with the following properties:*

1. *For all* $(x', v', r') \in U_j \times V_j \times \mathbb{R}_+$ *the equivalence* $g_j(x', r'Lv') = 0 \Leftrightarrow r' = \rho_j^{x,v}(x', v')$ *holds true.*

2. *For all* $(x', v') \in U_j \times V_j$ *one has the gradient formula*

$$\nabla_x \rho_j^{x,v}(x', v') = -\frac{1}{\left\langle \nabla_z g_j\left(x', \rho_j^{x,v}(x', v') Lv'\right), Lv'\right\rangle} \nabla_x g_j\left(x', \rho_j^{x,v}(x', v') Lv'\right).$$

Lemma 2.31. *[100, Lem 3.1] Let* $x \in \mathbb{R}^n$ *be such that* $g^{\mathrm{m}}(x, 0) < 0$ *and let* $v \in F(x)$*. Then, introducing the index set* $J_F^{x,v} := \{j \in \{1, \ldots, k\} \mid v \in F_j(x)\}$*, the functions* $\rho_j^{x,v}$ *from Lemma 2.30 are well-defined for* $j \in J_F^{x,v}$ *on the neighbourhood* $\tilde{U} \times \tilde{V}$ *of* (x, v)*, where, with* U_j, V_j *from Lemma 2.30,*

$$\tilde{U} := \cap_{j \in J_F} U_j, \quad \tilde{V} := \cap_{j \in J_F} V_j.$$

Moreover, there exist neighbourhoods $U \subseteq \tilde{U}$ *of* x *and* $V \subseteq \tilde{V}$ *of* v *with the following properties:*

1. *For all* $(x', v', r') \in U \times V \times \mathbb{R}_+$ *the equivalence* $g^{\mathrm{m}}(x', r'Lv') = 0 \Leftrightarrow r' = \rho^{x,v}(x', v')$ *holds true, where* $\rho^{x,v} : \tilde{U} \times \tilde{V} \to \mathbb{R}_+$ *is defined as*

$$\rho^{x,v}(x', v') := \min_{j \in J_F^{x,v}} \rho_j^{x,v}(x', v') \quad \forall (x', v') \in \tilde{U} \times \tilde{V}.$$

2. *For all* $(x', v') \in U \times V$*, the partial Clarke-sub-differential of* $\rho^{x,v}$ *(w.r.t.* x*) is given by*

$$\partial_x^c \rho^{x,v}(x', v') = \operatorname{conv}\left\{\nabla_x \rho_j^{x,v}(x', v') : j \in \mathcal{J}^{x,v}(x', v')\right\}.$$

where $\operatorname{conv}(A)$ *stands for the convex hull of a set* A *and* $\mathcal{J}^{x,v}(x', v') := \left\{j \in J_F^{x,v} \mid \rho_j^{x,v}(x', v') = \rho^{x,v}(x', v')\right\}$.

A difficulty that we must pay attention is when the set $M(x)$ is unbounded at a target point $\bar{x}$. Such condition may lead a non-Lipschitzian behaviour of φ [92, Example 2.3]. To handle unboundedness we can assume additional conditions to control the growth of $\nabla_x g$

for large values of z. Now we define the $\theta_{\mathcal{R}}$-growth condition that makes a relation with the underlying random vector ξ through its radial component $\mathcal{R}$.

Definition 2.32. *[92, Def. 2.1] Let $\theta_{\mathcal{R}} : \mathbb{R}_+ \to \mathbb{R}_+$ be an increasing mapping such that for any $\delta > 0$ the following condition holds:*

$$\lim_{r\to\infty} f_{\mathcal{R}}(r) r \theta_{\mathcal{R}}(\delta r) = 0,$$

where $f_{\mathcal{R}}$ is the density of $\mathcal{R}$. Let $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ be a differentiable function. We say that h satisfies the $\theta_{\mathcal{R}}$-growth condition at $\bar{x}$ if for some $\delta_1, C > 0$ and neighbourhood U of $\bar{x}$ it holds that

$$\|\nabla_x h(x,z)\| \leq \delta_1 \theta_{\mathcal{R}}(\|z\|),$$

for all $x \in U$ and z such that $\|z\| \geq C$

Now we are able to provide a differentiability result for the probability function φ:

Theorem 2.33. *[94, Thm. 1] Assume that:*

- *The mapping g is continuously differentiable and convex in the second argument*
- *The random vector ξ is elliptically symmetrically distributed with positive definite covariance-like matrix Σ and continuous generator.*

Let the following conditions be satisfied at some fixed $\bar{x} \in \mathbb{R}^n$:

1. *There exists $\varepsilon > 0$, such that $g_j(\bar{x}, 0) < -\varepsilon$, for $j = 1, \ldots, k$*
2. *g_j satisfies the $\theta_{\mathcal{R}}$-growth condition at $\bar{x}$ (Definition 2.32) for all $j = 1, \ldots, k$.*

Then, φ given by $x \mapsto \varphi(x) := \mathbb{P}\left[g(x,\xi) \leq 0\right]$ is locally Lipschitz continuous on a neighbourhood U of $\bar{x}$ and it holds that

$$\partial^c \varphi(x) \subseteq \int\limits_{v \in \mathcal{D}om(\rho(x,.))} conv\left\{ -\frac{f_{\mathcal{R}}\big(\rho(x,v)\big)}{\big\langle \nabla_z g_j\big(x, \rho(x,v)Lv\big), Lv\big\rangle} \nabla_x g_j\big(x, \rho(x,v)Lv\big) \mid j \in \hat{\mathcal{J}}(x,v) \right\} d\mu_\zeta(v) \tag{2.15}$$

for all $x \in U$*. Here, for any* $v \in \mathrm{Dom}\big(\rho(x,.)\big)$*,*

$$\hat{\mathcal{J}}(x,v) := \big\{ j \in \{1,\dots,k\} \mid g_j\big(x, \rho(x,v)Lv\big) = 0 \big\}$$

refers to the active index set.

The integral in (2.15) is to be understood as the set of integrals over all measurable selections of the set valued integrand [94]. As we can see, Theorem 2.33 only achieve a statement on the locally Lipschitzian nature of the probability function and an outer estimate of the subdiffrential, but still provides a good path to establish the continuous differentiability of φ. To achieve such condition we will need a constraint qualification for g. For any $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^m$, we denote by

$$\mathcal{I}(x,z) := \{ j \in \{1,\dots,k\} \mid g_j(x,z) = 0 \}$$

the active index set of g at (x,z). We say that the inequality system $g(x,z) \leq 0$ satisfies the *Rank-2-Constraint Qualification* (*R2CQ*) at $x \in \mathbb{R}^n$ if

$$\mathrm{rank}\,\{\nabla_z g_j(x,z), \nabla_z g_i(x,z)\} = 2 \quad \forall i,j \in \mathcal{I}(x,z), i \neq j, \quad \forall z \in \mathbb{R}^m : g(x,z) \leq 0. \quad (2.16)$$

Under this constraint qualification condition we can finally provide the differentiability of φ.

Corollary 2.34. *[94, Cor. 1] In addition to the assumptions of Theorem 2.3, suppose that* (2.16) *is satisfied at* $\bar{x}$*. Then,* φ *is Fréchet differentiable at* $\bar{x}$ *and the gradient formula*

$$\nabla\varphi(\bar{x}) = -\int\limits_{v \in \mathcal{D}\mathrm{om}(\rho(\bar{x},.)), \#\hat{\mathcal{J}}(\bar{x},v)=1} \frac{f_{\mathcal{R}}\big(\rho(\bar{x},v)\big)}{\big\langle \nabla_z g_{j(v)}\big(\bar{x}, \rho(\bar{x},v)Lv\big), Lv \big\rangle} \nabla_x g_{j(v)}\big(\bar{x}, \rho(\bar{x},v)Lv\big) d\mu_\zeta(v) \quad (2.17)$$

holds true. Here $j(v)$ *is the unique index* $j \in \{1,\dots,k\}$ *satisfying* $g_j\big(\bar{x}, \rho(\bar{x},v)Lv\big) = 0$*. If* (2.16) *is satisfied locally around* $\bar{x}$*, then,* φ *is continuously differentiable on an appropriate neighbourhood of* $\bar{x}$*.*

Considering a spherical-radial decomposition of the non-degenerate Gaussian random vector ξ, [102, Thm. 3] proves that $\varphi(x) = \mathbb{P}[g(x,\xi) \leq 0]$ is twice continuously differentiable on a neighbourhood U of $\bar{x}$ (see [102, Sec. 4] for examples of chi-squared, lognormal and Student random vectors). The function g is assumed to be twice continuously differentiable, convex with respect to the second argument and it satisfies the first and second order exponential growth conditions [102, Assumption 1] on the Hessian $\nabla^2 g(x)$ at $\bar{x}$. Combining this result and the compactness of the feasible set $X \subset \mathbb{R}^n$ (see Chapter 1), we have that $\nabla^2\varphi(x)$ is bounded by the following standard result on analysis.

Theorem 2.35. *Let K be a nonempty subset of $\mathbb{R}^n$, where $n \geq 1$. If K is compact, then every continuous real-valued function defined on K is bounded.*

Consequently, the next result ensures that $\nabla\varphi(x)$ is Lipschitz continuous in X.

Lemma 2.36. *[64, Lem. 1.2.2] The function $\varphi : X \subset \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable in X and satisfies $\|\nabla\varphi(x) - \nabla\varphi(y)\| \leq L\|x - y\|$, for a constant $L > 0$, if and only if*

$$\|\nabla^2\varphi(x)\| \leq L, \quad \forall x \in \mathbb{R}^n. \tag{2.18}$$

In other words, we have that every probability function that is of class $\mathcal{C}^2$ with bounded Hessian (2.18) has Lipschitz continuous gradient.

2.3 Copulæ

We now focus on the particular case of distribution functions called copulæ, whose domain is the m-dimensional unit box $\mathbb{I}^m := [0,1]^m$. Multivariate distribution functions contain two types of information: the description of the marginal behaviour and the dependence structure. The last one is where copulæ take part, they allow us to represent the dependencies between multivariate distributions just on the basis of its one-dimensional marginals. In other words, we can construct any multivariate distribution function by separately specifying the marginal distributions and the copula.

In this section we will present some definitions, properties, examples, the classical Sklar's theorem, which states the existence of a copula associated to the distribution function, and a classification of copulæ in families, according to some characteristics.

2.3.1 Definition and properties

Definition 2.37. *[29, Def. 1.3.1] For every $m \geq 2$, an m-dimensional copula (an m-copula) is an m-dimensional distribution function concentrated on $\mathbb{I}^m$ whose univariate marginals are uniformly distributed on $\mathbb{I}$. The set of m-copulæ is denoted by $\mathcal{C}_m$.*

An immediate consequence from Theorem 2.8 is: to each copula C there exists a random vector $\mathbf{U} = (U_1, \ldots, U_m)$ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ such that C is the joint distribution function of $\mathbf{U}$. Such a probabilistic characterization allows the introduction of the following three fundamental examples of copulæ, where we illustrate their graph and the t-level set (defined below) in the Figures 2.4, 2.5 and 2.6.

Definition 2.38. *[29, Def. 1.8.2] Let $C \in \mathcal{C}_m$ and let $t \in \mathbb{I}$. The t-level set is the set of all points $u \in \mathbb{I}^m$ such that $C(u) = t$. It is defined by $L_C^t = \{u \in \mathbb{I}^m : C(u) = t\}$.*

Notice that, for every $t \in \mathbb{I}$, all the points of type $(t, 1, \ldots, 1), (1, t, 1, \ldots, 1), \ldots,$ $(1, 1, \ldots, 1, t)$ belong to L_C^t because the uniform distributions of the marginals.

Example 2.39. *[29, e.g. 1.3.3] (The copula M_m) Let U be a random variable defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Suppose that U is uniformly distributed on $\mathbb{I}$. Consider the random vector $\boldsymbol{U} = (U, \ldots, U)$. Then, for every $u \in \mathbb{I}^m$,*

$$\mathbb{P}[\boldsymbol{U} \leq u] = \mathbb{P}[U \leq \min\{u_1 \ldots, u_m\}] = \min\{u_1, \ldots, u_m\}.$$

Thus the distribution function given, for every $u \in \mathbb{I}^m$, by

$$M_m(u_1, \ldots, u_m) := \min\{u_1, \ldots, u_m\}$$

is a copula, which will be called the comonotonicity copula. The graph and the t-level set are represented in Figure 2.4 for $m = 2$.

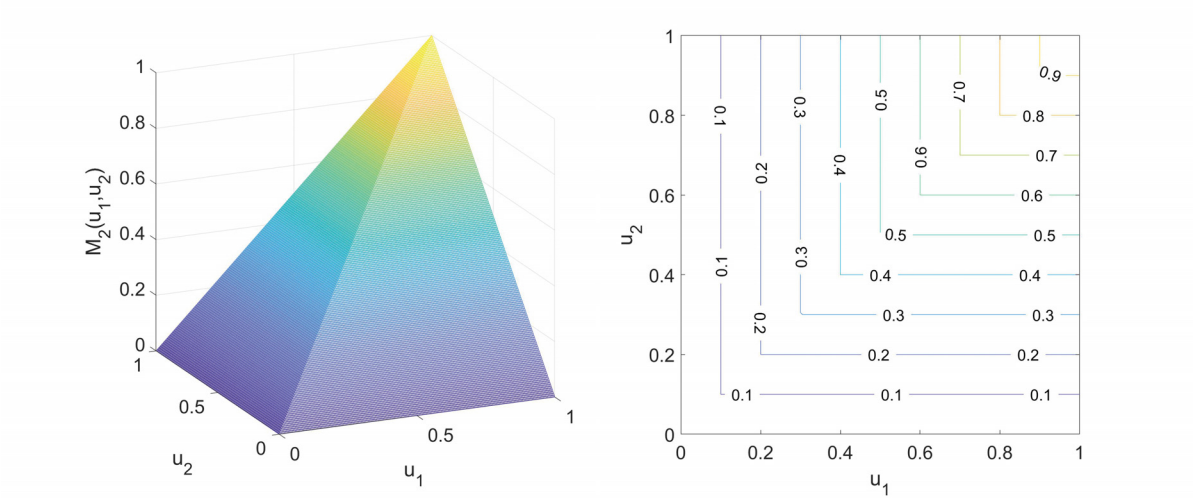


Figure 2.4: 3-d graph (left) and the t-level set (right) of the comonotonicity copula for $m = 2$.

Example 2.40. *[29, e.g. 1.3.4] (The copula* Π_m*) Let* $U_1, \ldots, U_m$ *be independent random variables defined on the probability space* $(\Omega, \mathcal{F}, \mathbb{P})$. *Suppose that each* U_i *is uniformly distributed on* $\mathbb{I}$. *Consider the random vector* $\boldsymbol{U} = (U_1, \ldots, U_m)$. *Then, for every* $u \in \mathbb{I}^m$,

$$\mathbb{P}\left[\boldsymbol{U} \leq u\right] = \mathbb{P}\left[U_1 \leq u_1\right] \cdots \mathbb{P}\left[U_m \leq u_m\right] = \prod_{j=1}^{m} u_j.$$

Thus the distribution function given, for every $u \in \mathbb{I}^m$, *by*

$$\Pi_m(u_1, \ldots, u_m) := \prod_{j=1}^{m} u_j$$

is a copula, which will be called independence copula. The graph and the t-level set are represented in Figure 2.5 for $m = 2$.

Example 2.41. *[29, e.g. 1.3.5] (The copula* W_2*) Let* U *be a random variable defined on the probability space* $(\Omega, \mathcal{F}, \mathbb{P})$. *Suppose that* U *is uniformly distributed on* $\mathbb{I}$. *Consider the*

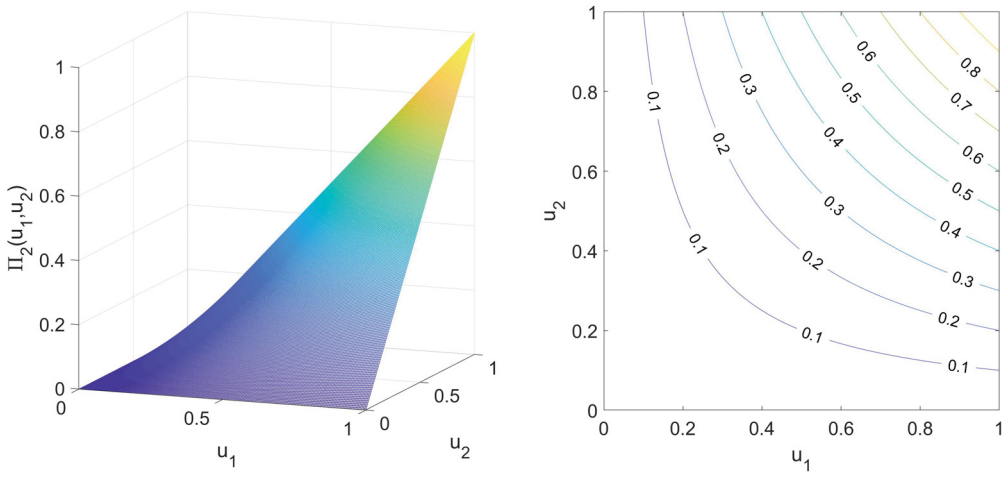


Figure 2.5: 3-d graph (left) and the t-level set (right) of the independence copula for $m = 2$.

random vector $\boldsymbol{U} = (U, 1-U)$. *Then, for every* $u \in \mathbb{I}^2$,

$$\mathbb{P}\left[\boldsymbol{U} \leq u\right] = \mathbb{P}\left[U \leq u_1, 1-U \leq u_2\right] = \max\{0, u_1 + u_2 - 1\}.$$

Thus the distribution function given, for every $u \in \mathbb{I}^2$, *by*

$$W_2(u_1, u_2) := \max\{0, u_1 + u_2 - 1\}$$

is a copula, which will be called the countermonotonicity copula. The graph and the t*-level set are represented in Figure 2.6 for* $m = 2$.

Now we introduce the standard partial order among real-valued functions in the space of copulæ, which gives us a result that provides upper and lower bounds in $\mathcal{C}_m$ with respect to the given order.

Definition 2.42. *[29, Def. 1.7.1] Let* $C, C' \in \mathcal{C}_m$. C *is less than* C' *in the pointwise order, and one writes* $C \leq C'$, *if, and only if,* $C(u) \leq C'(u)$ *for every* $u \in \mathbb{I}^m$.

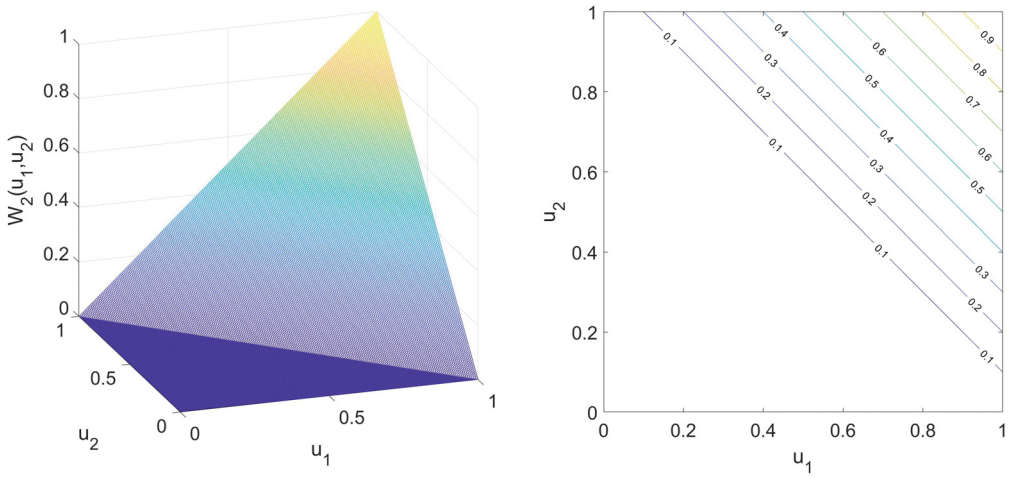


Figure 2.6: 3-d graph (left) and the t-level set (right) of the countermonotonicity copula for $m = 2$.

To the next result, consider the function $W_m : \mathbb{I}^m \to \mathbb{I}$ defined by

$$W_m(u) := \max\left\{0, \sum_{j=1}^{m} u_j - (m-1)\right\}.$$

Theorem 2.43. *[29, Thm. 1.7.3] For every m-copula C and for every point $u = (u_1, \ldots, u_m) \in \mathbb{I}^m$, one has*

$$W_m(u) \leq C(u) \leq M_m(u), \tag{2.19}$$

where M_m is defined in Example 2.39.

Proof. Let the copula C be the distribution function of a random vector $\mathbf{U}$ that is defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We will prove each side of the inequality (2.19).

- $C(u) \leq M_m(u)$

  For every index $j \in \{1, \ldots, m\}$ and for every $u \in \mathbb{I}^m$, one has

$$\bigcap_{k=1}^{m} \{U_k \leq u_k\} \subseteq \{U_j \leq u_j\}$$

which implies that

$$C(u) = \mathbb{P}\left[\bigcap_{k=1}^{m} \{U_k \leq u_k\}\right] \leq \min_{j\in\{1,\ldots,m\}} \mathbb{P}\left[U_j \leq u_j\right] = M_m(u).$$

- $C(u) \geq W_m(u)$

  Analogously, one has

$$\begin{aligned} C(u) &= \mathbb{P}\left[\bigcap_{j=1}^{m} \{U_j \leq u_j\}\right] = 1 - \mathbb{P}\left[\bigcup_{j=1}^{m} \{U_j > u_j\}\right] \\ &\geq 1 - \sum_{j=1}^{m} \mathbb{P}\left[U_j > u_j\right] = 1 - \sum_{j=1}^{m} (1 - u_j) = \sum_{j=1}^{m} u_j - (m-1). \end{aligned}$$

  Since C takes positive values, it follows that $C(u) \geq W_m(u)$.

□

The functions W_m and M_m are called the *lower* and *upper Hoeffding-Fréchet bounds*, respectively. A family of copulæ that includes W_m, Π_m and M_m is said to be *comprehensive*.

Besides the probabilistic interpretation of copulæ, they can be characterized in terms of the analytical properties of the distribution functions, as a consequence of Theorem 2.8.

Theorem 2.44. *[29, Thm. 1.4.1] A function $C : [0,1]^m \to [0,1]$ is called a copula if the following conditions hold:*

(a) $C(u_1, \ldots, u_m) = 0$ if $u_j = 0$ for at least one index $j \in \{1, \ldots, m\}$;

(b) When all the arguments of C are equal to 1, except possibly for the j-th one, then

$$C(1, \ldots, 1, u_j, 1, \ldots, 1) = u_j;$$

(c) C is quasi-monotone on $[0,1]^m$.

Properties *(a)* and *(b)* together are called the *boundary conditions* of a m-copula. Property (c) means that the C-volume of any box in $[0,1]^m$ is nonnegative (a property satisfied by probability functions) and it is also found in the literature as m-increasing (see Theorem 2.8 (b)). This can be interpreted in such a way that the copula C is increasing in each variable, i.e., for every $j \in \{1,\ldots,m\}$ and for all $u_1,\ldots,u_{j-1},u_{j+1},\ldots,u_m$ in $\mathbb{I}$, $t \mapsto C(u_1,\ldots,u_{j-1},u_j(t),u_{j+1},\ldots,u_m)$ is increasing.

From the above definitions and results, a basic way to prove that a function $C:\mathbb{I}^m \to \mathbb{I}$ is a copula is to verify its definition, i.e., finding a suitable probabilistic model whose distribution function is concentrated on $\mathbb{I}^m$ and has uniform marginals. A second way consists in proving that the three properties of Theorem 2.44 are satisfied. However, this latter strategy is usually complex to demonstrate in high dimensions. In order to simplify the calculations of the m-increasing property, we define the F-volume of a function F, which will be useful to prove the next results.

Definition 2.45. *[29, Def. 1.2.10] Let A be a rectangle in $\bar{\mathbb{R}}^m$, where $\bar{\mathbb{R}}$ stands for the extended real line $[-\infty,+\infty]$. For a function $F:A\to\mathbb{R}$, the F-volume V_F of $(a,b]\subseteq A$ is defined by*

$$V_F\big((a,b]\big) := \sum_{v\in \mathtt{ver}\left((a,b]\right)} sign(v)F(v),$$

where

$$sign(v) = \begin{cases} 1, & \text{if } v_j = a_j \text{ for an even number of indices,} \\ -1, & \text{if } v_j = a_j \text{ for an odd number of indices,} \end{cases}$$

and $ver\big((a,b]\big) = \{a_1,b_1\}\times\cdots\times\{a_m,b_m\}$ is the set of the vertices of $(a,b]$.

Definition 2.46. *[29, Def. 1.2.11] Let A be a rectangle in $\bar{\mathbb{R}}^d$. A function $H:\bar{\mathbb{R}}^d\to\mathbb{R}$ is m-increasing if the H-volume V_H of every rectangle $(a,b]$ is positive, i.e., $V_H\big((a,b]\big)\geq 0$.*

By definition 2.46, the function W_m is not a copula for $m \geq 3$. In [63, Exc. 2.36] is possible see that for the rectangle determined by the m-dimensional vectors $\mathbf{1/2} = [1/2,\ldots,1/2]$ and $\mathbf{1} = [1,\ldots,1]$, the volume $V_{W_m}\big([\mathbf{1/2},\mathbf{1}]\big) = 1-(m/2)$ is negative if

$m \geq 3$, and then Theorem 2.44 *(c)* does not hold. On the other hand, for each $u \in \mathbb{I}^m$ there exists $C_u \in \mathcal{C}_m$, which depends on u, such that $C_u(u) = W_m(u)$ [29, Thm. 4.1.7].

Example 2.47. *If the domain of F is $\mathbb{R}^2$, then F is also said to be supermodular. In such a case, $V_F\big((a,b]\big)$, where $a = \{a_1, a_2\}$ and $b = \{b_1, b_2\}$, is written explicitly as*

$$\begin{aligned} V_F\big((a,b]\big) &= \sum_{v \in \mathtt{ver}\big((a,b]\big)} sign(v)F(v) \\ &= sign((a_1,a_2))F(a_1,a_2) + sign((a_1,b_2))F(a_1,b_2) + \\ &\quad + sign((b_1,a_2))F(b_1,a_2) + sign((b_1,b_2))F(b_1,b_2) \\ &= F(a_1,a_2) - F(a_1,b_2) + F(b_1,a_2) - F(b_1,b_2). \end{aligned}$$

In the following, if ξ is a random vector with distribution function F, then $V_F\big([a,b]\big) = \mathbb{P}\big[\xi \in [a,b]\big]$. Obviously, if F is continuous, $V_F\big((a,b]\big) = \mathbb{P}\big[\xi \in [a,b]\big]$ for all $a, b \in \mathbb{R}^m$ with $a \leq b$. Next lemma provides some properties of the F-volume V_F of a function F, which will be important to prove the convexity of the set of copulæ $\mathcal{C}_m$.

Lemma 2.48. *[29, Lem. 1.4.4] Let $F, G : \mathbb{I}^m \to \mathbb{I}$ be two functions. Let $(a,b]$ be a m-box in $\mathbb{I}^m$. Then:*

(a) $V_{F+G}\big((a,b]\big) = V_F\big((a,b]\big) + V_G\big((a,b]\big)$;

(b) $V_{\alpha F}\big((a,b]\big) = \alpha V_F\big((a,b]\big)$ *for every* $\alpha > 0$;

(c) if $(a,b] = \cup_{j \in \mathcal{J}} B_j$, where $\mathcal{J}$ has finite cardinality and all B_j's are left open m-boxes whose interiors are disjoint, then

$$V_F\big((a,b]\big) = \sum_{j \in \mathcal{J}} V_F(B_j).$$

Theorem 2.49. *[29, Thm. 1.4.5] The set $\mathcal{C}_m$ is a convex set, i.e., for all $\alpha \in \mathbb{I}$ and C_0 and C_1 in $\mathcal{C}_m$, $C = \alpha C_0 + (1-\alpha)C_1$ is in $\mathcal{C}_m$.*

Proof. Let $C_0, C_1 \in \mathcal{C}_m$, $\alpha \in \mathbb{I}$ and let $C = \alpha C_0 + (1-\alpha)C_1$ be a convex combination of C_0 and C_1. Its easily proved that the univariate marginals of C are uniformly distributed on $\mathbb{I}$. Moreover, for every rectangle $(a, b] \subseteq \mathbb{R}^m$, using Lemma 2.48 yields

$$\begin{aligned} V_C\big((a,b]\big) &= V_{\alpha C_0+(1-\alpha)C_1}\big((a,b]\big) \\ &= V_{\alpha C_0}\big((a,b]\big) + V_{(1-\alpha)C_1}\big((a,b]\big) \\ &= \alpha V_{C_0}\big((a,b]\big) + (1-\alpha)V_{C_1}\big((a,b]\big), \end{aligned}$$

which is the desired assertion. □

Example 2.50. *[29, e.g. 1.4.6] Consider the case $m = 2$ and let α and β be in $\mathbb{I}$ with $\alpha + \beta \leq 1$. Then, in view of the convexity of $\mathcal{C}_2$, $C_{\alpha,\beta} : \mathbb{I}^2 \to \mathbb{I}$ defined by*

$$C^{\textbf{\textit{Fre}}}_{\alpha,\beta}(u_1, u_2) := \alpha M_2(u_1, u_2) + (1-\alpha-\beta)\Pi_2(u_1, u_2) + \beta W_2(u_1, u_2)$$

is a copula. As the parameters α and β vary in $\mathbb{I}$ subject to the restriction $\alpha + \beta \leq 1$, the copula varies in a family of copulæ known as the Fréchet copulæ.

The next result gives us a Lipschitz condition of a m-copula.

Theorem 2.51. *[29, Thm. 1.5.1] A m-copula C satisfies the following condition, for all $u, v \in \mathbb{I}^m$:*

$$|C(u_1, \ldots, u_m) - C(v_1, \ldots, v_m)| \leq \sum_{j=1}^{m} |u_j - v_j|. \tag{2.20}$$

Proof. Let C be the distribution function of a random vector ξ defined $(\Omega, \mathcal{F}, \mathbb{P})$ and let $F_1, \ldots, F_m$ be its univariate marginals. Then, for every $j \in \{1, \ldots, m\}$, for $t_j < t'_j$ and for every $(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_m) \in \mathbb{R}^{m-1}$,

$$\begin{aligned} &C(x_1, \ldots, x_{j-1}, t'_j, x_{j+1}, \ldots, x_m) - C(x_1, \ldots, x_{j-1}, t_j, x_{j+1}, \ldots, x_m) \\ &= \mathbb{P}\left[\xi_1 \leq x_1, \ldots, \xi_j \leq t'_j, \ldots, \xi_m \leq x_m\right] - \mathbb{P}\left[\xi_1 \leq x_1, \ldots, \xi_j \leq t_j, \ldots, \xi_m \leq x_m\right] \\ &= \mathbb{P}\left[\xi_1 \leq x_1, \ldots, t < \xi_j \leq t'_j, \ldots, \xi_m \leq x_m\right] = F_j(t') - F_j(t) \\ &= t' - t. \end{aligned}$$

Note that the last equality holds because the univariate marginals $F_1, \ldots, F_m$ of C follow a uniform distribution. By the triangular inequality, we have that

$$\begin{aligned} |C(u) - C(v)| &\leq |C(u) - C(v_1, u_2, \ldots, u_m)| \\ &\quad + |C(v_1, u_2, \ldots, u_m) - C(v_1, v_2, u_3, \ldots, u_m)| \\ &\quad + |C(v_1, v_2, u_3 \ldots, u_m) - C(v_1, v_2, v_3, u_4, \ldots, u_m)| \\ &\quad + \cdots + |C(v_1, \ldots, v_{m-1}, u_m) - C(v)| \\ &\leq \sum_{j=1}^{m} |F_j(u_j) - F_j(v_j)| = \sum_{j=1}^{m} |u_j - v_j|. \end{aligned}$$

□

It is possible to show that there is no better possible constant of the Lipschitz condition in (2.20) than 1. In other words, does not exist a constant $\alpha < 1$ such that

$$|C(u_1, \ldots, u_m) - C(v_1, \ldots, v_m)| \leq \alpha \sum_{j=1}^{m} |u_j - v_j| = \alpha \parallel u - v \parallel_1 .$$

We refer to the inequality (2.20) as the Lipschitz condition with constant 1, or simply the *1-Lipschitz condition*. One can also say that every copula $C \in \mathcal{C}_m$ is 1-Lipschitz continuous with respect to the $\ell_1(m)$-norm. In particular, every m-copula C is uniformly continuous on $\mathbb{I}^m$. Now, analogously to Definition 2.10, we introduce a stronger version of continuity in $\mathcal{C}_m$.

Definition 2.52. *[29, Def. 1.5.4]A copula $C \in \mathcal{C}_m$ is absolutely continuous if it can be expressed in the form*

$$C(u) = \int_{[0,u]} c(t)\ dt$$

for a suitable integrable function $c : \mathbb{I}^m \to \mathbb{R}_+$.

As we have seen before, the function c is called the *density* of C. An obvious example is the copula Π_m, which is absolutely continuous with density $c = 1$. Since a copula has uniform marginals, the density of an absolutely continuous copula can be characterized by

means of the following property: for every $t \in \mathbb{I}$ and for every $j \in \{1, \ldots, m\}$,

$$\int_0^1 \cdots \int_0^1 \underbrace{\int_0^t}_{\text{j-th integral}} \int_0^1 \cdots \int_0^1 c(u)\, \mathrm{d}u_m \ldots \mathrm{d}u_{j+1}\, \mathrm{d}u_j\, \mathrm{d}u_{j-1} \ldots \mathrm{d}u_1 = t. \tag{2.21}$$

Among all the definitions and properties over copulæ that we have seen, they also have a special meaning when dealing with joint distribution functions, which is the key point to work with copulæ. Every multivariate distribution function of a random vector ξ contains two types of information: the description of the marginal behaviour, which means the probabilistic knowledge of the single components of the random vector, and the dependence structure.

Considering that we know the behaviour of the single components of a random vector ξ in terms of their univariate distribution functions, by means of the the next result a suitable multivariate model can be constructed.

Theorem 2.53. *[29, Thm. 2.1.1] Let $F_1, \ldots, F_m$ be univariate distribution functions and let C be any m-copula. Then the function $F : \mathbb{R}^m \to \mathbb{I}$ defined, for every point $x = (x_1, \ldots, x_m) \in \mathbb{R}^m$, by*

$$F(x_1, \ldots, x_m) = C\big(F_1(x_1), \ldots, F_m(x_m)\big), \tag{2.22}$$

is an m-dimensional distribution function with margins given by $F_1, \ldots, F_m$.

This result suggests us an approach to build a multivariate distribution. Defining the marginal distributions, where we can give attention to univariate distributions with different natures, a copula may be chosen in such a way that the marginals are linked to a common model. Since some families of multivariate distribution require that the marginals are in the same family, this recipe seems quite promising.

Now, for our purposes, we present the result of utmost importance, which is the Sklar's theorem. This theorem allows us to connect the probability law of any multivariate random vector to its marginal distributions through a copula.

Theorem 2.54. *[29, Thm. 2.2.1](Sklar's theorem) Let $\xi \in \mathbb{R}^m$ be a random vector defined in the probability space $(\Omega, \mathcal{F}, \mathbb{P})$,*

$$F(y) := \mathbb{P}\left[\xi_1 \leq y_1, \ldots, \xi_m \leq y_m\right]$$

be the joint distribution function of ξ and $F_j(y_j) = \mathbb{P}\left[\xi_j \leq y_j\right]$, $j = 1, \ldots, m$, be its marginals. Then, there exists a copula C_ξ such that, for every point $y = (y_1, \ldots, y_m) \in \mathbb{R}^m$,

$$F(y) = C_\xi\big(F_1(y_1), \ldots, F_m(y_m)\big). \tag{2.23}$$

If the marginals $F_1, \ldots, F_m$ are continuous, then the copula C_ξ is uniquely defined.

In essence, Sklar's theorem states that a multivariate distribution function may be expressed as a composition of a copula and its univariate marginals. The existence of a unique copula of a m-dimensional distribution function whose marginals $F_1, \ldots, F_m$ are continuous is ensured in the following Lemma.

Lemma 2.55. *[29, Lem. 2.2.3] Under the assumptions of Theorem 2.54, if $F_1, \ldots, F_m$ are continuous, then there exists a unique copula C associated with ξ that is the distribution function of the random vector $(F_1 \circ \xi_1, \ldots, F_m \circ \xi_m)$. It is determined, for every $u \in \mathbb{I}^m$, via the formula*

$$C(\mathbf{u}) = F\left(F_1^{(-1)}(u_1), \ldots, F_m^{(-1)}(u_m)\right), \tag{2.24}$$

where, for $j \in \{1, \ldots, m\}$, $F_j^{(-1)}$ is the quasi-inverse of F_j.

Considering some particular properties of the copulæ, we will classify them into families.

2.3.2 Families of copulæ

The objective of this section is to present some of the several families of copulæ that have appeared in the literature with interesting theoretical properties and applications. We begin introducing the Archimedean family, which is used in the numerical experiments in Chapter 4, and then follows the Fréchet, EFGM and Elliptical families of copulæ.

Archimedean copulæ

Archimedean copulæ are parametrized via a one-dimensional function, which is defined below.

Definition 2.56. *Given a real parameter θ, a function $\psi_\theta : [0,1] \to [0,\infty)$ is said to be a (copula) generator if it is convex, continuous, strictly decreasing on $[0,t_0]$, where $t_0 = \inf\{t > 0 : \psi_\theta(t) = 0\}$, and $\psi_\theta(1) = 0$.*

The inverse of the generator ψ_θ is written as ψ_θ^{-1}, and its pseudo-inverse $\psi_\theta^{[-1]}$ is defined by

$$\psi_\theta^{[-1]}(t) := \begin{cases} \psi_\theta^{-1}(t) & \text{if } \ 0 \le t \le \psi_\theta(0) \\ 0 & \text{if } \ \psi_\theta(0) \le t \le +\infty. \end{cases}$$

The following definition introduces m-dimensional *Archimedean copulæ* for $m \ge 2$.

Definition 2.57. *A copula C is called Archimedean if it has the representation*

$$C(u_1,\dots,u_m) = \psi_\theta^{[-1]}\Big(\psi_\theta(u_1) + \dots + \psi_\theta(u_m)\Big), \tag{2.25}$$

where $\psi_\theta : [0,1] \to [0,\infty)$ is a generator function.

Remark 2.58. *In the literature we can find another equivalent definition for the generator and the Archimedean copula. For example, in [29], $\psi_\theta^{[-1]}$ and ψ_θ are replaced by φ_θ and $\varphi_\theta^{(-1)}$, respectively.*

With reference to the generator, the copula (2.25) is denoted by C_ψ. When ψ is strictly decreasing in the whole interval $[0,1]$, its pseudo-inverse $\psi^{[-1]}$ equals its inverse, $\psi^{[-1]} = \psi^{-1}$, and the copula C_ψ is said to be *strict*.

Example 2.59. *The copula Π_2 is Archimedean: take $\psi(t) = -\log t$; since $\lim_{t\to 0}\psi(t) = +\infty$, $\psi(t) > 0$ for every $t \in [0,1)$ and $\psi(1) = 0$, ψ is strict; then $\psi(t)^{-1} = e^{-t}$ and*

$$\psi^{-1}\big(\psi(u) + \psi(v)\big) = \exp\big(-(-\log u - \log v)\big) = uv = \Pi_2(u,v).$$

We saw that an Archimedean copula depends on a generator. An important task is which properties that a generator has to enjoy in order that the function C_ψ defined by (2.25) is a m-copula. This question will be addressed and answered in Theorem 2.62 below via the following preliminary definition.

Definition 2.60. *[29, Def. 6.5.5] A function $f : (a, b) \to \mathbb{R}$ is called m-monotone in (a, b), where $-\infty \leq a < b \leq +\infty$ and $m \geq 2$ if*

- *it is differentiable up to order $m - 2$;*
- *for every $x \in (a, b)$, its derivatives satisfy*

$$(-1)^k f^{(k)}(x) \geq 0$$

  for $k = 0, \ldots, m - 2$;

- *$(-1)^{m-2} f^{m-2}$ is decreasing and convex in (a, b).*

Moreover, if f has derivatives of every order in (a, b) and if

$$(-1)^k f^{(k)}(x) \geq 0,$$

for every $x \in (a, b)$ and for every $k \in \mathbb{Z}_+$,f is said to be completely monotone.

Definition 2.61. *[29, Def. 6.5.6] Let $I \subseteq \mathbb{R}$ be an interval. A function $f : I \to \mathbb{R}$ is said to be m-monotone (respectively, completely monotone) on I, with $m \in \mathbb{N}$, if it is continuous on I and if its restriction to the interior I° of I is m-monotone (respectively, completely monotone).*

Theorem 2.62. *[29, Thm. 6.5.7] Let $\psi : [0, +\infty] \to \mathbb{I}$ be a generator. Then the following statements are equivalent:*

(a) ψ is m-monotone on $[0, +\infty)$;

(b) the function $C_\psi : \mathbb{I}^m \to \mathbb{I}$ defined by (2.25) is a m-copula.

Now we list some families of Archimedean copulæ, showing their expression and its Archimedean generator [29, e.g. 6.5.16 - 6.5.19] .

i. *Gumbel-Hougaard copulæ:*

The Archimedean generator, and its inverse, of this family are given by

$$\psi_\theta(t) = \big(-\log(t)\big)^\theta \quad \text{and} \quad \psi_\theta^{(-1)}(t) = \exp(-t^{1/\theta}),\ \theta \in [1,\infty). \tag{2.26}$$

The standard expression for members of this family of m-copulæ is

$$C_\theta^{\mathbf{GH}}(u) = \exp\left(-\left(\sum_{i=1}^{m}(-\log(u_i))^\theta\right)^{1/\theta}\right), \tag{2.27}$$

where $\theta \geq 1$. For $\theta = 1$ one obtains the independence copula as a special case, and the limit of $C_\theta^{\mathbf{GH}}$ for $\theta \to +\infty$ is the comonotonicity copula M_m. Each member of this class is absolutely continuous.

The expression (2.27) is obtained by applying the generator (2.26) in (2.25):

$$\begin{aligned} C_\theta^{\mathbf{GH}}(u) &= \psi_\theta^{(-1)}\Big(\psi_\theta(u_1) + \ldots + \psi_\theta(u_m)\Big) \\ &= \psi_\theta\Big((-\log(u_1))^\theta + \ldots + (-\log(u_m))^\theta\Big) \\ &= \exp\Big(-\big((-\log(u_1))^\theta + \ldots + (-\log(u_m))^\theta\big)^{1/\theta}\Big). \end{aligned}$$

ii. *Mardia-Takashi-Clayton copulæ:*

The Archimedean generator, and its inverse, of this family is given by

$$\psi_\theta(t) = \frac{1}{\theta}(t^{-\theta} - 1) \quad \text{and} \quad \psi_\theta^{(-1)}(t) = (\max\{1 + \theta t, 0\})^{-1/\theta},\ \theta \in [-1,\infty)\backslash\{0\}.$$

The standard expression for members of this family of m-copulæ is

$$C_\theta^{\mathbf{MTC}}(u) = \max\left\{\left(\sum_{i=1}^{m} u_i^{-\theta} - (m-1)\right)^{-1/\theta}, 0\right\},$$

where $\theta \geq -1/(m-1)$, $\theta \neq 0$. The limiting case $\theta \to 0$ corresponds to the independence copula.

iii. *Frank copulæ:*

The Archimedean generator, and its inverse, of this family is given by

$$\psi_\theta(t) = -\log\left(\frac{e^{-\theta t}-1}{e^{-\theta}-1}\right) \quad \text{and} \quad \psi_\theta^{(-1)}(t) = \frac{1}{\theta}\log\left(1-(1-e^{-\theta})e^{-t}\right),\ \theta \in \mathbb{R}\backslash\{0\}.$$

The standard expression for members of this Frank family of m-copulæ is

$$C_\theta^{\mathbf{Frank}}(u) = -\frac{1}{\theta}\log\left(1+\frac{\prod_{i=1}^m(e^{-\theta u_i}-1)}{(e^{-\theta}-1)^{m-1}}\right),$$

where $\theta > 0$. The limiting case $\theta = 0$ corresponds to Π_m. For $m = 2$, the parameter θ can be extended also to the case $\theta < 0$.

iv. *Ali-Mikhail-Haq copulæ:*

The Archimedean generator, and its inverse, of this family is given by

$$\psi_\theta(t) = \log\left(\frac{1-\theta}{t}+\theta\right) \quad \text{and} \quad \psi_\theta^{(-1)}(t) = \frac{1-\theta}{e^t-\theta},\ \theta \in [-1,1).$$

The standard expression for members of the Ali-Mikhail-Haq (AMH) family of 2-copulæ is

$$C_\theta^{\mathbf{AMH}}(u,v) = -\frac{uv}{1-\theta(1-u)(1-v)},$$

where $\theta \in [-1,1]$. For $\theta = 0$ one has $C_0 = \Pi_2$.

v. *Joe's copulæ:*

The Archimedean generator, and its inverse, of this family is given by

$$\psi_\theta(t) = -\log\left(1-(1-t)^\theta\right) \quad \text{and} \quad \psi_\theta^{(-1)}(t) = 1-(1-e^{-t})^{1/\theta},\ \theta \geq 1.$$

The standard expression for members of the Joe's family of m-copulæ is

$$C_\theta^{\mathbf{Joe}}(u) = 1 - \left(1 - \prod_{i=1}^{m}\Big(1-(1-u_i)^\theta\Big)\right)^{1/\theta},$$

where $\theta \geq 1$.

Fréchet copulæ

This family of copulæ came from studies about the upper and lower bounds in the class of distribution functions with fixed margins, as given in (2.19). Then, a convex combination of these functions in the Fréchet class creates a parametric family. This two-parameter family may be represented in the form

$$C_{\alpha,\beta}^{\mathbf{Fre}}(u_1,u_2) := \alpha M_2(u_1,u_2) + (1-\alpha-\beta)\Pi_2(u_1,u_2) + \beta W_2(u_1,u_2),$$

where α and β are in $\mathbb{I}$ with $\alpha+\beta \leq 1$.

Since the Fréchet lower bound is not a copula for $m \geq 3$, as we have already mentioned, this family cannot be fully extended to the higher dimensional case. A possible m-dimensional extension of its subclass describing positive dependence is given by, for every $\alpha \in \mathbb{I}$,

$$C_\alpha^{\mathbf{Fre}}(\mathbf{u}) := \alpha M_m(\mathbf{u}) + (1-\alpha)\Pi_m(\mathbf{u}). \tag{2.28}$$

EFGM copulæ

A bivatirate *Eyraud-Farlie-Gumbel-Morgenstern* (EFGM) copula has the following expression,

$$C_\alpha^{\mathbf{EFGM}}(u_1,u_2) := u_1u_2(1+\alpha(1-u_1)(1-u_2)),$$

with $\alpha \in [-1,1]$. Now, consider the higher dimensional extension to $m \geq 3$. Let $\mathcal{I}$ be the class of all subsets of $\{1,\ldots,m\}$ having at least 2 elements, so that $\mathcal{I}$ contains $2^m - m - 1$

elements. To each $S \in \mathcal{I}$, we associate a real number α_S, with the convention that, when $S = \{i_1, \ldots, i_k\}$, $\alpha_S = \alpha_{i_1,\ldots,i_k}$. An EFGM m-copula can be defined in the following form:

$$C_\alpha^{\mathbf{EFGM}}(\mathbf{u}) = \prod_{i=1}^{m} u_i \Big(1 + \sum_{S\in\mathcal{I}} \alpha_S \prod_{j\in S}(1-u_j)\Big), \tag{2.29}$$

for suitable values of the α_S's.

Elliptical copulæ

This section is also based on the references [11, 66, 101]. The elliptical copulæ are obtained by applying the inverse transformation (2.24) to multivariate elliptical distributions.

Definition 2.63. *[29, Def. 6.7.1] An elliptical copula is any copula that can be obtained from an elliptical distribution using the inversion method of equation* (2.24).

Example 2.64. *[29, e.g. 6.7.2] The Gaussian copula is the copula of an elliptical random variable ξ that follows a Gaussian distribution, i.e.,*

$$\xi \overset{m}{=} \boldsymbol{A}\boldsymbol{Z},$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times k}$, $\Sigma := \boldsymbol{A}\boldsymbol{A}^T \in \mathbb{R}^{m\times m}$ is the covariance matrix, $rank(\Sigma) = k \leqslant m$ and $\boldsymbol{Z}$ is an m-dimensional random vector whose independent components have univariate standard Gaussian law. We write $\xi \sim N_m(\mu, \Sigma)$.

The bivariate Gaussian copula is given by

$$C_\rho^{\boldsymbol{Ga}}(u,v) = \int_{-\infty}^{\Phi^{-1}(u)} \mathrm{d}s \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{s^2 - 2\rho st + t^2}{2\,(1-\rho^2)}\right) \mathrm{d}t,$$

where ρ is in $(-1,1)$, and Φ^{-1} denotes the inverse of the standard Gaussian distribution $N(0,1)$.

Example 2.65. *[29, e.g. 6.7.3] The Student's t-copula is the copula of an elliptical random*

vector ξ that follows a multivariate Student's t-distribution, i.e.,

$$\xi \overset{m}{=} \mu + \Sigma^{1/2}\sqrt{W}\mathbf{Z},$$

where $\mathbf{Z} \sim N_m(\boldsymbol{0}, \boldsymbol{I}_m)$ is a Gaussian distribution, $\Sigma := \Sigma^{1/2}\Sigma^{1/2}$ is positive definite. Moreover, W and $\mathbf{Z}$ are independent, and W follows an inverse Gamma distribution with parameters $(\nu/2, \nu/2)$.

The bivariate Student's t-copula is given by

$$C^{\boldsymbol{t}}_{\rho,\nu}(u, v) = \mathbf{t}_{\rho,\nu}\left(t^{-1}_{\nu}(u), t^{-1}_{\nu}(v)\right),$$

where ρ is in $(-1, 1)$, and $v > 1$, while $\mathbf{t}_{\rho,\nu}$ is the bivariate Student t-distribution with zero mean, the correlation matrix having off-diagonal element ρ, and ν degrees of freedom, while t^{-1}_{ν} denotes the inverse of the standard t-distribution. The Student t-copula becomes a Gaussian copula in the limit $\nu \to \infty$.

Chapter 3

Trust-region algorithm with copula-based models

In this chapter, which is the core of our publication [10], we present a derivative-free trust-region algorithm to solve probability maximization problem (1.2). Our method builds upon [15], but differs from the latter in the definition of the model and the iterates. While [15] computes iterates as stationary points of quadratic constrained programs, our approach defines iterates as approximate stationary points of nonlinear optimization problems arising from approximating the difficult probability function with a simple copula-based model. The definition of the model, the algorithm, the global convergence analysis and auxiliary results ensuring some conditions and assumptions are presented in the sequence.

3.1 Copula-based model

This section discusses the construction of the copula-based model. It is separated in two parts, in the first one we apply the Sklar's theorem to the probability function φ in (1.2), ensuring the existence of a copula that coincides with φ when it is composed with the function g and the univariate marginals of φ. In the second one we define the model, which is given by a linear combination of copulæ such that the coefficients of such combination

are a solution of a simple least-squares quadratic problem.

3.1.1 Application of Sklar's theorem

In the context of the optimization problem (1.2), Theorem 2.54 asserts that there exists a copula $C_\xi : [0,1]^m \to [0,1]$ such that the objective function $\varphi(x) = \mathbb{P}\left[\xi \leq g(x)\right]$ can be represented as the composition of the mapping $g : \mathcal{O} \to \mathbb{R}^m$, the marginals $F_j : \mathbb{R} \to [0,1]$, $j = 1, \ldots, m$, and the copula C_ξ:

$$\begin{aligned} \varphi(x) &= C_\xi\big(\mathbb{P}\left[\xi_1 \leq g_1(x)\right], \ldots, \mathbb{P}\left[\xi_m \leq g_m(x)\right]\big) \\ &= C_\xi\Big(F_1\big(g_1(x)\big), \ldots, F_m\big(g_m(x)\big)\Big). \end{aligned} \tag{3.1}$$

Observe that this theorem is not constructive, it "only" ensures the existence of a copula associated with the cumulative distribution function F. In most of the practical cases, a copula providing the equality (2.23) is unknown. Estimating a suitable one is a non-trivial task that has been receiving much attention in the last few years [29, 63]. Instead of finding a single $C \in \mathcal{C}_m$ fitting φ, we consider a dictionary $\mathcal{D}_r$ with $r \in \mathbb{N}$ copulæ of class $\mathcal{C}^1$ and Lipschitz continuous gradient to locally approximate φ:

$$\mathcal{D}_r := \left\{ C_i, i = 1, \ldots, r \;\middle|\; \begin{array}{l} C_i \text{ is a copula of class } \mathcal{C}^1 \text{ with} \\ \text{Lipschitz continuous gradient} \end{array} \right\} \subset \mathcal{C}_m. \tag{3.2}$$

The composition, similar to (3.1), of a copula $C_i \in \mathcal{D}_r$, the marginals F_j, $j = 1, \ldots, m$, and the mapping g, will be denoted by C_i^F, i.e.,

$$C_i^F(x) := C_i\Big(F_1\big(g_1(x)\big), \ldots, F_m\big(g_m(x)\big)\Big). \tag{3.3}$$

The central idea in this work is to iteratively find a vector $\lambda^k \in \mathbb{R}^r$ such that the model $\sum_{i=1}^r \lambda_i^k C_i^F$ approximates φ locally. The term "locally" is related to a (trust) region around a specific point, which we detail in the next section.

3.1.2 The model

Let $\mathcal{D}_r$ be a dictionary defined as in (3.2) and $\{x^0, \ldots, x^k\} \subset X$ be the set of points issued by the algorithm up to iteration k. The best candidate to solve (1.2) among these points is denoted by *stability center* $\hat{x}^k$. Furthermore, at iteration k, we define G_k (not necessarily in X) a set with finitely many points at which the function φ has been evaluated. Our approach defines a copula-based model $\mathcal{M}^k : \mathcal{O} \to \mathbb{R}$ of φ as

$$\mathcal{M}^k(x) := \sum_{i=1}^{r} \lambda_i^k C_i^F(x), \tag{3.4a}$$

where C_i^F is the composition given in (3.3) for each $C_i \in \mathcal{D}_r$ and the coefficients λ_i^k, $i = 1, \ldots, r$, solve the quadratic programming problem

$$\begin{aligned} \text{minimize} \quad & \sum_{x^j \in G_k} \left(\sum_{i=1}^{r} \lambda_i C_i^F(x^j) - \varphi(x^j) \right)^2 \\ \text{subject to} \quad & \sum_{i=1}^{r} \lambda_i C_i^F(\hat{x}^k) = \varphi(\hat{x}^k) \\ & \lambda \in \Lambda. \end{aligned} \tag{3.4b}$$

In this notation, Λ is either a large enough box in $\mathbb{R}^r$ or the simplex $\Lambda = \left\{\lambda \in \mathbb{R}^r_+ : \sum_{i=1}^r \lambda_i = 1\right\}$. In both cases the model $\mathcal{M}^k$, given in (3.4), reflects the variational properties of the involved functions. For instance, $\mathcal{M}^k$ is continuously differentiable with Lipschitz continuous gradient on X provided the marginal functions F_j share the same property on $g_j(X)$, $j = 1, \ldots, m$.

More details on the possible choices for the set G_k will be given later on in Subsection 3.2.3. For now, we care to mention that G_k plays an important role in the convergence analysis of our method. Customarily, G_k is constructed around $\hat{x}^k$ to ensure that the model $\mathcal{M}^k$ approximates φ well enough (in terms of hypothesis A3 below) in a neighborhood of the stability center. This is a standard requirement in DFO methods, and here we use a known procedure to construct/update G_k: in our numerical experiments, we employ Algorithm 4.2 from [111]. As it will be detailed in Subsection 3.2.3, such procedure yields

G_k with $n+1$ well-chosen points, and ensures that the optimal value of (3.4b) is zero provided Λ is a large enough box in $\mathbb{R}^r$ and the copula dictionary is sufficiently rich, i.e., the model $\mathcal{M}^k$ interpolates φ at the points in G_k. We also investigate a more economical rule (in terms of function evaluation) that proves efficient in practice: at every iteration k, we choose a small set G_k containing $\{x^0, \ldots, x^k\}$ and let Λ be the simplex in $\mathbb{R}^r$. In this case, the model in (3.4) is a convex combination of copulæ and thus a copula itself. We will see in Subsection 3.2.3 that convergence analysis requires more stringent assumptions on the dictionary $\mathcal{D}_r$. Observe that if (3.4b) is infeasible, then the dictionary $\mathcal{D}_r$ is poor: the correlations of the joint probability function can not be represented by the copulæ in $\mathcal{D}_r$. It is then necessary to enlarge the dictionary, either by including new families of copulæ or by considering different parameters for C_i in $\mathcal{D}_r$, when C_i is an Archimedean copula, for example.

3.1.3 The algorithm

Our approach considers a zero-order oracle to compute φ at a given point, where no first-order information (gradient) is required. The next iterate x^{k+1} is defined as an approximate stationary point of the trust-region subproblem

$$\max_{x \in X} \ \mathcal{M}^k(x) \quad \text{s.t.} \quad \|x - \hat{x}^k\|_\diamond \leq \Delta_k, \tag{3.5}$$

where $\Delta_k > 0$ is the radius defining the trust region $B(\hat{x}^k, \Delta_k) := \{x \in \mathbb{R}^n : \|x - \hat{x}^k\|_\diamond \leq \Delta_k\}$, and $\|\cdot\|_\diamond$ is a given norm[1]. The more points of G_k are in $X \cap B(\hat{x}^k, \Delta_k)$, the more $\mathcal{M}^k$ can be trusted in this region. Since $\mathcal{M}^k$ is of class $\mathcal{C}^1$, small radii yield regions where the model approximates well the objective function φ. Without additional assumptions, (3.5) is a nonconvex optimization problem. Hence, solving it globally with optimality guarantees is a difficult task. For our purpose, similar to [85], it is enough to compute x^{k+1}

[1]We used the ℓ_∞-norm in our implementations.

as an approximate stationary point satisfying the *efficiency condition*

$$\mathcal{M}^k(x^{k+1}) - \mathcal{M}^k(\hat{x}^k) \geq c_1 \pi_k^2 \min\left\{ \frac{\pi_k^2}{\beta}, \Delta_k, 1 \right\}, \tag{3.6}$$

where $c_1 > 0$ and $\beta \geq 1$ are constants independent of k, and π_k is the stationarity measure given by

$$\pi_k = \left\| \texttt{Proj}_X\Big(\hat{x}^k + \nabla \mathcal{M}^k(\hat{x}^k)\Big) - \hat{x}^k \right\|. \tag{3.7}$$

In this notation, $\|\cdot\|$ denotes the ℓ_2-norm and $\texttt{Proj}_X$ stands for the orthogonal projection onto X, which exists and is unique because X is a nonempty compact convex set. We recall that $x^* \in X$ is stationary for the original problem (1.2) if

$$\left\| \texttt{Proj}_X\big(x^* + \nabla \varphi\,(x^*)\,\big) - x^* \right\| = 0,$$

which suggests us to use π_k as a stopping test for the method: under appropriate assumptions (see A1-A3 in Section 3.2), $\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k) = O(\Delta_k)$ and, thus, π_k is indeed a stationary measure provided that Δ_k is small enough.

The efficiency condition (3.6) is inspired by the classical Cauchy step condition in [65, Lem. 4.5] on trust-region algorithms for solving unconstrained problems, and in [19, Thm. 10.1] for the derivative-free case. Similar conditions also appear in different contexts in [16], in the design of filter methods for nonlinear programming in [40, 67], and for bound-constrained nonlinear optimization without derivatives in [87]. Condition (3.6) is attainable and less demanding than finding an exact stationary point for (3.5). In the Appendix, we detail how to adapt the algorithm proposed in [85] for computing x^{k+1} satisfying (3.6).

Observe that when $\Delta_k, \pi_k > 0$, condition (3.6) implies $\mathcal{M}^k(x^{k+1}) - \mathcal{M}^k(\hat{x}^k) > 0$ and the ratio γ_k between the actual and predicted increase

$$\gamma_k = \frac{\varphi(x^{k+1}) - \varphi(\hat{x}^k)}{\mathcal{M}^k(x^{k+1}) - \mathcal{M}^k(\hat{x}^k)} \tag{3.8}$$

is well defined. As $\mathcal{M}^k(\hat{x}^k) = \varphi(\hat{x}^k)$ by the construction of the model in (3.4), if $\gamma_k > 0$

then x^{k+1} is a better candidate than $\hat{x}^k$ to solve (1.2). This suggests a strategy for updating the stability center, as detailed in Algorithm 1.

Algorithm 1. Derivative-free trust-region algorithm with copula-based models

Input : $\hat{x}^0 \in X$, a dictionary $\mathcal{D}_r$ of copulæ as in (3.2) and parameters:
$0 < \alpha$, $0 \leqslant \eta < \eta_1 \leqslant \eta_2$, $0 < \tau_1 < 1 \leqslant \tau_2$, $0 \leq \texttt{tol} < \Delta_0 < \Delta_{\max}$

Set $x^0 = \hat{x}^0$, $G_0 = \{x^0\}$
for $k = 0, 1, 2, \ldots$ **do**
  if (3.4b) *is feasible* **then**
    Solve problem (3.4b), let λ^k be one of its solutions and set $\mathcal{M}^k(x) = \sum_{i=1}^{r} \lambda_i^k C_i^F(x)$
  else
    Stop: the copulæ in the dictionary $\mathcal{D}_r$ can not approximate well φ
  end if
  Compute π_k as in (3.7)
  if $\pi_k \leq \texttt{tol}$ *and* $\Delta_k \leq \texttt{tol}$ **then**
    Stop: return $\hat{x}^k$ and $\varphi(\hat{x}^k)$
  end if
  if $\Delta_k \leq \alpha\pi_k$ **then**
    Find an approximate solution x^{k+1} of (3.5) satisfying (3.6) and set γ_k by (3.8)
    if $\gamma_k > \eta$ **then**
      $\hat{x}^{k+1} = x^{k+1}$
    else
      $\hat{x}^{k+1} = \hat{x}^k$
    end if
    if $\gamma_k > \eta_1$ **then**
      if $\gamma_k > \eta_2$ *and* $\|x^{k+1} - \hat{x}^k\|_\diamond = \Delta_k$ **then**
        $\Delta_{k+1} = \min\{\tau_2\Delta_k, \Delta_{\max}\}$
      else
        $\Delta_{k+1} = \Delta_k$
      end if
    else
      $\Delta_{k+1} = \tau_1\Delta_k$
    end if
  else
    $\Delta_{k+1} = \tau_1\Delta_k$, $\hat{x}^{k+1} = \hat{x}^k$ and $x^{k+1} = \hat{x}^k$
  end if
  Choose finitely many points to create a set $Y_{k+1} \subset B(\hat{x}^{k+1}, \Delta_{k+1})$
  Set $G_{k+1} \subset G_k \cup Y_{k+1} \cup \{x^{k+1}\}$ according to a given rule
end for

Note that Algorithm 1 stops unsuccessfully when the copulæ in the dictionary $\mathcal{D}_r$ can not approximate the objective function, i.e., when the constraints of the problem (3.4b) are not satisfied. In this case, the quality of the dictionary needs to be improved. This can be done by enlarging $\mathcal{D}_r$ either by considering different parameters for the considered copulæ, or by adding new ones.

If $\Delta_k \leq \alpha\pi_k$, then the algorithm follows the general lines of classical trust-region methods that involve the following parameters: η to update the stability center and the pair η_1, η_2 to update (increase or decrease) the trust-region by factors τ_1 and τ_2, respectively. The choice of these parameters is largely discussed in the literature [16], and their values are commonly set around 0, 0.2, 0.6, 0.5 and 2, respectively. These are the values we used in our numerical experiments. When $\Delta_k > \alpha\pi_k$, the trust-region radius is decreased and the stability center is kept as is. Regardless whether this inequality is verified, the algorithm updates the set of points G_k, and thus the model, to ensure that a small value of π_k reflects on approximate stationarity of $\hat{x}^k$ to the original problem. We will discuss in Section 3.2.3 strategies for choosing points on line 31 and a rule for updating G_k so that a key hypothesis (see Assumption A3 below) for the convergence analysis of Algorithm 1 is satisfied.

3.2 Convergence analysis

We now rely on [15] to analyze Algorithm 1. Throughout this section we assume that `tol` $= 0$, the algorithm generates infinite sequences $\{x^k\} \subset X$, $\{\lambda^k\} \subset \Lambda$, and the following hypotheses hold:

A1. The objective function φ is differentiable on $\mathcal{O}$ and its gradient $\nabla\varphi$ is Lipschitz continuous with constant $\kappa_\varphi > 0$ on $X \subset \mathcal{O}$.

A2. The marginals F_j, constraint functions g_j, $j = 1, \ldots, m$, and copulæ $C_i \in \mathcal{D}_r$, $i = 1, \ldots, r$, are of class $\mathcal{C}^1$ with Lipschitz continuous gradient on X.

A3. There exists a constant $c_2 > 0$ such that, for all $k \in \mathbb{N}$ and $x \in B(\hat{x}^k, \Delta_k)$,

$$|\varphi(x) - \mathcal{M}^k(x)| \leq c_2 \Delta_k^2.$$

As commented in Chapter 1 and stated in Section 2.2.2, Assumptions A1 and A2 hold by many probability distributions for PMPs, such as multivariate Gaussian distribution (Lemmas 2.20 and 2.21 and Theorems 2.22 and 2.23), distributions satisfying some growth conditions ([102, Thm. 3]), general distributions with fairly few assumptions (Theorem 2.17) and also all distributions and copulæ of class $\mathcal{C}^2$ on X, together with the assumptions on g and X (Theorem 2.35 and Lemma 2.36). In particular, the families of Archimedean copulæ described in Section 2.3.2 are of class $\mathcal{C}^2$ on a subset of X whose image is not so close to zero. It is interesting to note that Algorithm 1 works naturally in this subset since it maximizes the model $\mathcal{M}^k$ and, consequently, the copulæ in the dictionary. More specifically, another way to ensure that Archimedean copulae satisfy A2 consists in using a modeling trick as follows.

Remark. Recall that any joint probability distribution satisfies $\mathbb{P}[\xi \leq g(x)] \leq \mathbb{P}[\xi_i \leq g_i(x)] = F_i(g_i(x))$, for all $i = 1, \ldots, m$ and all $x \in \mathcal{O}$. Let $\tilde{x} \in X$ be an arbitrary feasible point producing a strictly positive probability, i.e. $\mathbb{P}[\xi \leq g(\tilde{x})] \geq \epsilon > 0$. Hence, for $i = 1, \ldots, m$, we have that $F_i(g_i(\tilde{x})) \geq \epsilon$, which gives $g_i(\tilde{x}) \geq F_i^{-1}(\epsilon)$, the ϵ-quantile of the uni-dimensional (marginal) distribution F_i. As any solution $\bar{x}$ of the PMP must satisfies $g_i(\bar{x}) \geq F_i^{-1}(\epsilon)$, replacing the feasible set X by $\tilde{X} := \{x \in X : g_i(x) \geq F_i^{-1}(\epsilon),\ i = 1 \ldots, n\}$ in the PMP does not change its solutions but ensures that $0 \notin C(\tilde{X})$ for all Archimedean copula C. As a result, over this new feasible (sub)set, Archimedean copulae are of class $\mathcal{C}^1$ having Lipschitz continuous gradients.

Assumption A3 is usual in DFO [19, 110] and states that the model has to properly represent φ near the current stability center. Note that if the *exact* copula C_ξ (associated with the probability function φ) in Theorem 2.54 is included in the dictionary $\mathcal{D}_r$ (or at least it belongs to the space spanned by the copulæ in the dictionary) then A3 holds for all $x \in \mathcal{O}$ due to the identity (3.1). We will come back to the subject of satisfying Assumption

A3 in Section 3.2.3.

Our assumption on the feasible set X ensures that $\{x^k\}$ is a bounded sequence. This is also the case for the sequence $\{\lambda^k\} \subset \Lambda$ of model's coefficients because Λ in (3.4) is either a box or the simplex in $\mathbb{R}^r$. Boundedness of $\{\lambda^k\}$ ensures that the model issued by (3.4) has Lipschitz continuous gradient with a constant independent of the iteration k (c.f. Lemma 3.1). In what follows, we present some consequences of these assumptions.

3.2.1 Assumptions A1-A3: what do they yield?

As a consequence of Assumption A2, we show that the copula-based model has Lipschitz (uniformly) continuous gradient on X.

Lemma 3.1. *Suppose that Assumption A2 holds. Then, there exists a Lipschitz constant $\kappa_{\mathcal{M}} > 0$, independent of k, such that, for all $x, y \in X$ and $k \in \mathbb{N}$,*

$$\|\nabla \mathcal{M}^k(x) - \nabla \mathcal{M}^k(y)\| \leq \kappa_{\mathcal{M}} \|x - y\|. \tag{3.9}$$

Proof. By the model's definition in (3.4) and the triangle inequality, we have, for all $x, y \in X$ and $k \in \mathbb{N}$,

$$\left\|\nabla \mathcal{M}^k(x) - \nabla \mathcal{M}^k(y)\right\| \leq \sum_{i=1}^{r} |\lambda_i^k| \left\|\nabla C_i^F(x) - \nabla C_i^F(y)\right\| \leq \kappa_\lambda \sum_{i=1}^{r} \left\|\nabla C_i^F(x) - \nabla C_i^F(y)\right\|,$$

where κ_λ is a constant bounding $\{|\lambda^k|\}$. By Assumption A2, the gradient of the function C_i^F, $i = 1, \ldots, r$, is Lipschitz continuous with constant, say, $\kappa_i^F > 0$. Consequently, $\left\|\nabla \mathcal{M}^k(x) - \nabla \mathcal{M}^k(y)\right\| \leq \kappa_\lambda \sum_{i=1}^{r} \kappa_i^F \|x - y\|$, and the proof follows by setting $\kappa_{\mathcal{M}} = \kappa_\lambda \sum_{i=1}^{r} \kappa_i^F$. □

The next lemma establishes an error bound on the model's gradient at the stability center.

Lemma 3.2. *Suppose that Assumptions A1 to A3 hold. Then, there exist constants $c_3, c_4 >$*

0 *such that, for all* $k \in \mathbb{N}$*,*

$$\|\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)\| \leq \min\{c_3\Delta_k, c_4\}. \tag{3.10}$$

Proof. Let $k \in \mathbb{N}$. If $\|\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)\| = 0$, then the result holds trivially. Otherwise, consider an arbitrary direction $d \in \mathbb{R}^n$ with $\|d\|_\diamond \leq \Delta_k$, i.e., $\hat{x}^k + d \in B(\hat{x}^k, \Delta_k)$. Note that

$$\begin{aligned}\big(\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)\big)^T d = &-\varphi(\hat{x}^k + d) + \varphi(\hat{x}^k) + \nabla\varphi(\hat{x}^k)^T d + \mathcal{M}^k(\hat{x}^k + d) - \varphi(\hat{x}^k) \\ &- \nabla\mathcal{M}^k(\hat{x}^k)^T d + \varphi(\hat{x}^k + d) - \mathcal{M}^k(\hat{x}^k + d).\end{aligned}$$

From the triangle inequality and the fact that $\varphi(\hat{x}^k) = \mathcal{M}^k(\hat{x}^k)$, we have

$$\begin{aligned}\Big|\big(\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)\big)^T d\Big| \leq &|\varphi(\hat{x}^k + d) - \varphi(\hat{x}^k) - \nabla\varphi(\hat{x}^k)^T d| + |\varphi(\hat{x}^k + d) \\ &- \mathcal{M}^k(\hat{x}^k + d)| + |\mathcal{M}^k(\hat{x}^k + d) - \mathcal{M}^k(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)^T d|.\end{aligned}$$

Recall that κ_φ and $\kappa_\mathcal{M}$ denote the Lipschitz constants of $\nabla\varphi$ and $\nabla\mathcal{M}^k$, respectively, over the set X (see (1.4) and (3.9)). Let $\bar{\kappa} \geq \max\{\kappa_\varphi, \kappa_\mathcal{M}\}$ be given. Then, [64, Lem. 1.2.3] yields

$$|\varphi(\hat{x}^k + d) - \varphi(\hat{x}^k) - \nabla\varphi(\hat{x}^k)^T d| \leq \frac{\bar{\kappa}}{2}\|d\|^2$$

and

$$|\mathcal{M}^k(\hat{x}^k + d) - \mathcal{M}^k(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)^T d| \leq \frac{\bar{\kappa}}{2}\|d\|^2.$$

Furthermore, Assumption A3 gives $|\varphi(\hat{x}^k + d) - \mathcal{M}^k(\hat{x}^k + d)| \leq c_2\Delta_k^2$ because $\|d\|_\diamond \leq \Delta_k$. Thus, we have shown that

$$\Big|\big(\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)\big)^T d\Big| \leq \frac{1}{2}\bar{\kappa}\|d\|^2 + c_2\Delta_k^2 + \frac{1}{2}\bar{\kappa}\|d\|^2 \leq \tilde{c}_2\Delta_k^2,$$

with $\tilde{c}_2 = b\bar{\kappa} + c_2$ and $b > 0$ a constant satisfying $\|\cdot\| \leq \sqrt{b}\|\cdot\|_\diamond$. The existence of constants $a, b > 0$ satisfying $\sqrt{a}\|\cdot\|_\diamond \leq \|\cdot\| \leq \sqrt{b}\|\cdot\|_\diamond$ is ensured by the equivalence of norms in $\mathbb{R}^n$.

So far, d was considered an arbitrary direction satisfying $\|d\|_\diamond \leq \Delta_k$. Now, we take the

particular direction $\tilde{d} = \Delta_k \dfrac{\left(\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)\right)}{\|\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)\|_\diamond}$ (note that $\|\tilde{d}\|_\diamond = \Delta_k$). Then,

$$\left|\left(\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)\right)^T \tilde{d}\right| = \Delta_k \frac{\left\|\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)\right\|^2}{\|\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)\|_\diamond} \geq \sqrt{a}\Delta_k \left\|\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)\right\|,$$

issuing

$$\sqrt{a}\Delta_k \left\|\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)\right\| \leq \tilde{c}_2\Delta_k^2.$$

By considering line 21 of Algorithm 1, taking $c_3 = \tilde{c}_2/\sqrt{a}$ and $c_4 = c_3\Delta_{\max}$ we conclude the proof. □

3.2.2 Global convergence of the algorithm

Some of the results below are similar to the ones presented in [15], differing, essentially, by the quadratic terms in the efficiency condition (3.6) and by the context of the maximization problem.

Lemma 3.3. *Suppose that there exists $\bar{k} \in \mathbb{N}$ such that $\Delta_k > \alpha\pi_k$, for all iteration $k \geq \bar{k}$. Then the sequences $\{\Delta_k\}$ and $\{\pi_k\}$ converge to zero.*

Proof. From line 29 of Algorithm 1, the radius is reduced by the factor $\tau_1 \in (0, 1)$ in each iteration $k \geq \bar{k}$, then $\lim_{k\to\infty} \pi_k \leq \dfrac{1}{\alpha} \lim_{k\to\infty} \Delta_k = 0$, completing the proof. □

The hypothesis of last lemma implies that from the iteration $\bar{k}$, the stability center does not change, i.e., $\hat{x}^k = \hat{x}^{\bar{k}}$, for all $k \geq \bar{k}$. Otherwise, if this hypothesis does not hold, there exist infinitely many iterations such that $\Delta_k \leq \alpha\pi_k$ and then the ratio γ_k given by (3.8) is well defined. The global convergence of the algorithm is ensured in both cases. To show that, let us define the set $\mathcal{S}$ of successful iterations and $\bar{\mathcal{S}}$ as the subset of $\mathcal{S}$ in which the trust-region radius does not decrease. More precisely,

$$\mathcal{S} = \{k \in \mathbb{N} \mid \Delta_k \leq \alpha\pi_k \text{ and } \gamma_k > \eta\} \quad \text{and} \quad \bar{\mathcal{S}} = \{k \in \mathbb{N} \mid \Delta_k \leq \alpha\pi_k \text{ and } \gamma_k > \eta_1\}. \tag{3.11}$$

As $\eta_1 > \eta$, $\bar{\mathcal{S}} \subset \mathcal{S}$. The next lemma asserts that if the trust-region radius is small enough, then the algorithm will perform a successful iteration in which the trust-region radius will not decrease, in the sense that $k \in \bar{\mathcal{S}}$.

Lemma 3.4. *Suppose that Assumptions A1 to A3 hold. Consider the constants c_1, β given in (3.6), c_2 defined in Assumption A3 and η_1 given in Algorithm 1.*

Set $c = c_2/c_1$ and let $\mathcal{K}$ be given by

$$\mathcal{K} = \left\{ k \in \mathbb{N} \mid \Delta_k \leq \min\left\{ \frac{\pi_k^2}{\beta}, \alpha\pi_k, \frac{(1-\eta_1)\,\pi_k^2}{c}, 1 \right\} \right\}. \tag{3.12}$$

Then it holds that $\mathcal{K} \subseteq \bar{\mathcal{S}}$.

Proof. Consider $k \in \mathcal{K}$. By the definitions of γ_k and the model $\mathcal{M}^k$, Assumption A3 and the fact that $x^{k+1} \in B(\hat{x}^k, \Delta_k)$,

$$\begin{aligned} |\gamma_k - 1| &= \left| \frac{\varphi(x^{k+1}) - \varphi(\hat{x}^k) - \left(\mathcal{M}^k(x^{k+1}) - \mathcal{M}^k(\hat{x}^k)\right)}{\mathcal{M}^k(x^{k+1}) - \mathcal{M}^k(\hat{x}^k)} \right| = \left| \frac{\varphi(x^{k+1}) - \mathcal{M}^k(x^{k+1})}{\mathcal{M}^k(x^{k+1}) - \mathcal{M}^k(\hat{x}^k)} \right| \\ &\leq \frac{c_2\Delta_k^2}{|\mathcal{M}^k(x^{k+1}) - \mathcal{M}^k(\hat{x}^k)|}. \end{aligned}$$

As $k \in \mathcal{K}$, $\Delta_k \leq \alpha\pi_k$ and, consequently, $\pi_k > 0$. It follows from (3.6) that

$$|\gamma_k - 1| \leq \frac{c_2\Delta_k^2}{c_1\pi_k^2 \min\left\{\frac{\pi_k^2}{\beta}, \Delta_k, 1\right\}} = \frac{c\Delta_k^2}{\pi_k^2 \min\left\{\frac{\pi_k^2}{\beta}, \Delta_k, 1\right\}}.$$

It follows from (3.12) that $\Delta_k = \min\left\{\frac{\pi_k^2}{\beta}, \Delta_k, 1\right\}$ and $\Delta_k \leq \frac{(1-\eta_1)\pi_k^2}{c} \Rightarrow \frac{c\Delta_k}{\pi_k^2} \leq 1 - \eta_1$. Therefore, $|\gamma_k - 1| \leq 1 - \eta_1$, which implies $\gamma_k \geq \eta_1 > \eta$ and consequently $k \in \bar{\mathcal{S}}$. This concludes the proof. □

Lemma 3.2 says that the smaller the radius Δ_k, the better the model approximates the objective function φ. Based on that, it is reasonable to expect that the sequence of trust-region radii converges to zero. This is ensured by the following lemma.

Lemma 3.5. *Suppose that Assumption A2 holds. Then the sequence* $\{\Delta_k\}$ *converges to zero.*

Proof. Assume, first, that the set $\bar{\mathcal{S}}$ defined in (3.11) is finite. Then, there exists $k_0 \in \mathbb{N}$ such that for all $k \geq k_0$, $\gamma_k \leq \eta_1$ or $\Delta_k > \alpha\pi_k$. By lines 26 and 29 of Algorithm 1, $\Delta_{k+1} = \tau_1\Delta_k$, for all $k \geq k_0$, with $0 < \tau_1 < 1$. Thus, $\{\Delta_k\}$ converges to zero. We assume henceforth $\bar{\mathcal{S}}$ is infinite. For any $k \in \bar{\mathcal{S}}$, using (3.6) we have

$$\begin{aligned} \varphi(x^{k+1}) - \varphi(\hat{x}^k) &\geq \eta_1 \left(\mathcal{M}^k(x^{k+1}) - \mathcal{M}^k(\hat{x}^k)\right) \\ &\geq \eta_1 c_1 \pi_k^2 \min\left\{\tfrac{\pi_k^2}{\beta}, \Delta_k, 1\right\}. \end{aligned}$$

By the definition of $\bar{\mathcal{S}}$, $\Delta_k \leq \alpha\pi_k$ and $\gamma_k > \eta_1 > \eta$. From line 15 of Algorithm 1, we have that $\hat{x}^{k+1} = x^{k+1}$, and hence

$$\varphi(\hat{x}^{k+1}) - \varphi(\hat{x}^k) \geq \eta_1 c_1 \frac{\Delta_k^2}{\alpha^2} \min\left\{\frac{\Delta_k^2}{\beta\alpha^2}, \Delta_k, 1\right\}.$$

Since $\{\varphi(\hat{x}^k)\}$ is a monotone nondecreasing sequence and bounded from above ($\varphi(x) \in [0,1]$, for all $x \in X$), the left-hand side of the above expression converges to zero, therefore:

$$\lim_{k \in \bar{\mathcal{S}}} \Delta_k = 0. \tag{3.13}$$

Consider the set $\mathcal{U} = \{k \in \mathbb{N} \mid k \notin \bar{\mathcal{S}}\}$. If $\mathcal{U}$ is finite, then by (3.13) we have that $\lim_{k\to\infty} \Delta_k = 0$. Now suppose that $\mathcal{U}$ is infinite. Consider $k \in \mathcal{U}$ and denote ℓ_k the latest index in $\bar{\mathcal{S}}$ before k. Then ℓ_k is well-defined for all large k and $\Delta_k \leq \tau_2\Delta_{\ell_k}$, which implies that

$$\lim_{k \in \mathcal{U}} \Delta_k \leq \tau_2 \lim_{k \in \mathcal{U}} \Delta_{\ell_k} = \tau_2 \lim_{\ell_k \in \bar{S}} \Delta_{\ell_k}.$$

By (3.13) it follows that $\lim_{k \in \mathcal{U}} \Delta_k = 0$ which completes the proof. □

The next lemma shows that the stationarity measure π_k in (3.7) has a subsequence that converges to zero.

Lemma 3.6. *Suppose that Assumptions A1 to A3 hold. Then* $\liminf_{k\to\infty} \pi_k = 0$.

Proof. The proof is by contradiction. Suppose that there exist a constant $\varepsilon > 0$ and an integer k_0 such that $\pi_k \geq \varepsilon$ for all $k \geq k_0$. Take $\tilde{\Delta} = \min\left\{\frac{\varepsilon^2}{\beta}, \alpha\varepsilon, \frac{(1-\eta_1)\varepsilon^2}{c}, 1\right\}$ where c is defined in Lemma 3.4, η_1 and $\alpha > 0$ are given in Algorithm 1. Consider $k \geq k_0$. If $\Delta_k \leq \tilde{\Delta}$, then $k \in \mathcal{K}$, where the latter is defined in (3.12). By Lemma 3.4, $k \in \bar{\mathcal{S}}$ and thus, by line 21 or 23 of the Algorithm 1, $\Delta_{k+1} \geq \Delta_k$. It follows that the trust-region radius can only decrease if $\Delta_k > \tilde{\Delta}$, and in this case, $\Delta_{k+1} = \tau_1\Delta_k > \tau_1\tilde{\Delta}$. Therefore, one can see that for all $k \geq k_0$, $\Delta_k \geq \min\left\{\tau_1\tilde{\Delta}, \Delta_{k_0}\right\}$, which contradicts Lemma 3.5 and concludes the proof. □

Assuming a sufficient increase on the objective function, by setting $\eta > 0$ in the algorithm, the next lemma ensures that not only there exists a subsequence of π_k converging to zero, but also the whole sequence converges.

Lemma 3.7. *Suppose that Assumptions A1 to A3 hold, and* $\eta > 0$. *Then* $\lim_{k\to\infty} \pi_k = 0$.

Proof. Suppose by contradiction that for some $\varepsilon > 0$ the set $\mathbb{N}' = \{k \in \mathbb{N} \mid \pi_k \geq \varepsilon\}$ is infinite. By Lemma 3.5, the sequence $\{\Delta_k\}$ converges to zero. Then, there exists $k_0 \in \mathbb{N}$ such that for all $k \geq k_0$,

$$\Delta_k \leq \min\left\{\frac{\varepsilon^2}{\beta}, \alpha\varepsilon, \frac{(1-\eta_1)\,\varepsilon^2}{c}, 1\right\} \tag{3.14}$$

where the constant c is given in Lemma 3.4 and α and η_1 are defined in Algorithm 1. It follows from definition of $\mathbb{N}'$ that, for all $k \in \mathbb{N}'$ with $k \geq k_0$,

$$\Delta_k \leq \min\left\{\frac{\pi_k^2}{\beta}, \alpha\pi_k, \frac{(1-\eta_1)\,\pi_k^2}{c}, 1\right\}. \tag{3.15}$$

Lemma 3.4 then ensures that $k \in \bar{\mathcal{S}} \subseteq \mathcal{S}$, for $k \geq k_0$, $k \in \mathbb{N}'$. Given $k \in \mathbb{N}'$ with $k \geq k_0$, consider ℓ_k the first index such that $\ell_k > k$ and $\pi_{\ell_k} \leq \varepsilon/2$. The existence of ℓ_k is ensured by Lemma 3.6. So, $\pi_k - \pi_{\ell_k} \geq \varepsilon/2$. Using the definition of π_k, the triangle inequality and

the contraction property of projections, we have that

$$\begin{aligned}
\tfrac{\varepsilon}{2} &\leq \left\|\texttt{Proj}_X\left(\hat{x}^k + \nabla\mathcal{M}^k(\hat{x}^k)\right) - \hat{x}^k\right\| - \left\|\texttt{Proj}_X\left(\hat{x}^{\ell_k} + \nabla\mathcal{M}^{\ell_k}(\hat{x}^{\ell_k})\right) - \hat{x}^{\ell_k}\right\| \\
&\leq \left\|\texttt{Proj}_X\left(\hat{x}^k + \nabla\mathcal{M}^k(\hat{x}^k)\right) - \hat{x}^k - \texttt{Proj}_X\left(\hat{x}^{\ell_k} + \nabla\mathcal{M}^{\ell_k}(\hat{x}^{\ell_k})\right) + \hat{x}^{\ell_k}\right\| \\
&\leq 2\left\|\hat{x}^k - \hat{x}^{\ell_k}\right\| + \left\|\nabla\mathcal{M}^k(\hat{x}^k) - \nabla\mathcal{M}^{\ell_k}(\hat{x}^{\ell_k})\right\| \\
&= 2\left\|\hat{x}^k - \hat{x}^{\ell_k}\right\| + \left\|\nabla\mathcal{M}^k(\hat{x}^k) - \nabla\varphi(\hat{x}^k) + \nabla\varphi(\hat{x}^k) - \nabla\varphi(\hat{x}^{\ell_k}) + \nabla\varphi(\hat{x}^{\ell_k}) - \nabla\mathcal{M}^{\ell_k}(\hat{x}^{\ell_k})\right\| \\
&\leq 2\left\|\hat{x}^k - \hat{x}^{\ell_k}\right\| + \left\|\nabla\mathcal{M}^k(\hat{x}^k) - \nabla\varphi(\hat{x}^k)\right\| + \left\|\nabla\varphi(\hat{x}^k) - \nabla\varphi(\hat{x}^{\ell_k})\right\| + \left\|\nabla\varphi(\hat{x}^{\ell_k}) - \nabla\mathcal{M}^{\ell_k}(\hat{x}^{\ell_k})\right\|.
\end{aligned}$$

So, using Lemma 3.2 twice and Assumption A1, the previous inequality can be written as

$$\frac{\varepsilon}{2} \leq (2 + \kappa_\varphi)\left\|\hat{x}^k - \hat{x}^{\ell_k}\right\| + c_3\left(\Delta_k + \Delta_{\ell_k}\right). \tag{3.16}$$

Consider $J_k = \{i \in \mathcal{S} \mid k \leq i < \ell_k\}$. Note that, by (3.15), $k \in \mathcal{S}$, so $J_k \neq \emptyset$. By (3.11), for all $i \in J_k$, $\hat{x}^{i+1} = x^{i+1}$ as a result of line 15 of Algorithm 1. Using this and the facts that $i \in \mathcal{S}$ and condition (3.6) holds, we conclude that

$$\begin{aligned}
\varphi(\hat{x}^{i+1}) - \varphi(\hat{x}^i) &\geq \eta\left(\mathcal{M}^i(\hat{x}^{i+1}) - \mathcal{M}^i\left(\hat{x}^i\right)\right) \\
&\geq \eta c_1 \pi_i^2 \min\left\{\tfrac{\pi_i^2}{\beta}, \Delta_i, 1\right\}.
\end{aligned}$$

By the definition of ℓ_k, we have that $\pi_i > \varepsilon/2$ for all $i \in J_k$. As $i \geq k$, by (3.14), $\Delta_i \leq \varepsilon^2/\beta$ and $\Delta_i \leq 1$. Therefore,

$$\frac{\Delta_i}{2} \leq \frac{\varepsilon^2}{2\beta} < \frac{2\pi_i^2}{\beta} \Rightarrow \frac{\Delta_i}{4} \leq \frac{\varepsilon^2}{4\beta} < \frac{\pi_i^2}{\beta}.$$

It follows that $\varphi(\hat{x}^{i+1}) - \varphi(\hat{x}^i) > \frac{\eta c_1 \varepsilon^2 \Delta_i}{16}$ and hence

$$\Delta_i < \frac{16}{\eta c_1 \varepsilon^2}\left(\varphi(\hat{x}^{i+1}) - \varphi(\hat{x}^i)\right). \tag{3.17}$$

On the other hand,

$$\left\|\hat{x}^k - \hat{x}^{\ell_k}\right\| \leq \sum_{i \in J_k}\left\|\hat{x}^i - \hat{x}^{i+1}\right\| \leq \sum_{i \in J_k}\Delta_i,$$

which combined with (3.17) provides $\left\|\hat{x}^k - \hat{x}^{\ell_k}\right\| < \frac{16}{\eta c_1 \varepsilon^2}\left(\varphi(\hat{x}^{\ell_k}) - \varphi(\hat{x}^k)\right)$. By the fact that the sequence $\{\varphi(\hat{x}_k)\}$ is bounded and it is monotone nondecreasing, $\varphi(\hat{x}^{\ell_k}) - \varphi(\hat{x}^k) \to 0$. Therefore the subsequence $\left\{\|\hat{x}^k - \hat{x}^{\ell_k}\|\right\}_{k\in\mathbb{N}'}$ converges to zero, which together with Lemma 3.5, contradicts (3.16) and completes the proof. □

The previous lemmas allow us to prove the following global convergence result: the sequence generated by Algorithm 1 has a stationary accumulation point and, in particular, when $\eta > 0$, any accumulation point of the sequence is stationary.

Theorem 3.8. *Suppose that Assumptions A1 to A3 hold. Then*

$$\liminf_{k\to\infty} \left\|\texttt{Proj}_X\left(\hat{x}^k + \nabla\varphi(\hat{x}^k)\right) - \hat{x}^k\right\| = 0.$$

In addition, if $\eta > 0$, then

$$\lim_{k\to\infty} \left\|\texttt{Proj}_X\left(\hat{x}^k + \nabla\varphi(\hat{x}^k)\right) - \hat{x}^k\right\| = 0.$$

Proof. Consider $k \in \mathbb{N}$ arbitrary. By the triangle inequality, it follows that

$$\begin{aligned}\left\|\texttt{Proj}_X\left(\hat{x}^k + \nabla\varphi(\hat{x}^k)\right) - \hat{x}^k\right\| \leq{}& \left\|\texttt{Proj}_X\left(\hat{x}^k + \nabla\varphi(\hat{x}^k)\right) - \texttt{Proj}_X\left(\hat{x}^k + \nabla\mathcal{M}^k(\hat{x}^k)\right)\right\| \\ &+ \left\|\texttt{Proj}_X\left(\hat{x}^k + \nabla\mathcal{M}^k(\hat{x}^k)\right) - \hat{x}^k\right\|.\end{aligned} \tag{3.18}$$

Applying the contraction property of projections and Lemma 3.2 to the first term on the right-hand side, we have $\left\|\texttt{Proj}_X\left(\hat{x}^k + \nabla\varphi(\hat{x}^k)\right) - \texttt{Proj}_X\left(\hat{x}^k + \nabla\mathcal{M}^k(\hat{x}^k)\right)\right\| \leq \left\|\nabla\varphi(\hat{x}^k) - \nabla\mathcal{M}^k(\hat{x}^k)\right\| \leq c_3\Delta_k$. From this and the definition of π_k, it follows from (3.18) that $\left\|\texttt{Proj}_X\left(\hat{x}^k + \nabla\varphi(\hat{x}^k)\right) - \hat{x}^k\right\| \leq c_3\Delta_k + \pi_k$. If the hypothesis of Lemma 3.3 holds, then the results follow from that lemma. Otherwise, applying Lemmas 3.5 and 3.6 we prove the first statement and the result for $\eta > 0$ follows from Lemmas 3.5 and 3.7. □

We care to mention that in the above analysis, the rule for choosing the set G_{k+1} on line 32 of Algorithm 1 plays no role. The reason is that Assumption A3 yields the necessary

mathematical results for ensuring convergence. This fact rises the question: how can we update sets G_{k+1} and Y_{k+1} to ensure A3?

3.2.3 Ensuring Assumption A3

In order to ensure Assumption A3, related to the quality of the model, we can focus on the dictionary or on the construction of the interpolation set Y_{k+1} on line 31 of Algorithm 1, as discussed below.

A rich dictionary of copulæ

Linear and quadratic models constructed by interpolation or regression, under some conditions, satisfy Assumption A3 as proved in [19]. In our case, the models are based on copulæ and thus A3 is expected to hold whenever $\{x^0, \ldots, x^k\} \subset G_k$ and the exact copula C_ξ associated with the probability function φ of Theorem 2.54 belongs to the space spanned by the copulæ in the dictionary, i.e., when

$$C_\xi \in \mathcal{C}_{\mathcal{D}_r} = \left\{ \sum_{i=1}^{r} \lambda_i C_i : \lambda \in \Lambda \right\}.$$

The intuition behind this claim follows from the fact that, under theses hypotheses, the optimal value of (3.4b) is zero, i.e., the model interpolates the points in G_k. Moreover, as G_k grows and the trust-regions shrinks, (3.4) yields a model that fits C_ξ (recall that in our setting C_ξ is unique due to Theorem 2.54). The choice $\Lambda = \left\{\lambda \in \mathbb{R}^r_+ : \sum_{i=1}^r \lambda_i = 1\right\}$ seems appropriate because in this case $\mathcal{M}^k$ is a copula for all $k = 0, 1, \ldots$ Note that requesting that $C_\xi \in \mathcal{C}_{\mathcal{D}_r}$ is a stringent assumption as C_ξ is usually unknown. However, from a practical point of view, such an assumption is sounder than the more frequent practice of replacing the probability function by an estimated copula. A way to try to satisfy Assumption A3 consists in considering a large and diversified dictionary, ideally including families of *comprehensive* copulæ (see the formal definition in Section 2), or at least containing copulæ yielding lower and upper bounds for the underlying probability

function.

Interpolation points

Instead of focusing on the dictionary of copulæ, we may concentrate on a rule for selecting the set of points Y_{k+1} on line 31 of Algorithm 1 and let $G_{k+1} \supset Y_{k+1}$ so that Assumption A3 is satisfied. This is the common practice in the DFO community [1, 19, 45, 111]. The next theorem states that whenever a general model (not necessarily quadratic or copula-based one) satisfies certain interpolation conditions, then error bounds on the model and its gradient are available. In particular, this ensures that Assumption A3 can be fulfilled.

Theorem 3.9. *[111, Thm. 2.3] Suppose that φ and $\mathcal{M}$ are continuously differentiable in $B(\hat{x}, \Delta)$ and that $\nabla\varphi$ and $\nabla\mathcal{M}$ are Lipschitz continuous in this $B(\hat{x}, \Delta)$. Consider a set of $n+1$ points $\hat{x} + y_i$ such that $y_1 = 0$, $\|y_i\| \leq \Delta$, for $i = 2, \ldots, n+1$, and $\|Y^{-1}\| \leq \frac{\Lambda_Y}{\Delta}$ for some constant $\Lambda_Y < \infty$, where Y is the square matrix $Y = \begin{bmatrix} y_2 & \cdots & y_{n+1} \end{bmatrix}$. If, for all $i = 1, \ldots, n+1$,*

$$\mathcal{M}(\hat{x} + y_i) = \varphi(\hat{x} + y_i),$$

then there exist constants γ_f and γ_g such that, for any $x \in B(\hat{x}, \Delta)$,

$$|\varphi(x) - \mathcal{M}(x)| \leq \gamma_f \Delta^2 \quad \textit{and} \quad \|\nabla\varphi(x) - \nabla\mathcal{M}(x)\| \leq \gamma_g \Delta.$$

Theorem 3.9 (whose proof can be found in [112, Thm. 4.1]) ensures Assumption A3 and a stronger result than Lemma 3.2, guaranteeing (3.10) in the whole trust region, not only in the stability center $\hat{x}$. These results imply that the model $\mathcal{M}$ is *fully linear* in the neighborhood $B(\hat{x}, \Delta)$ containing $n+1$ interpolation points, according to [19, Def. 6.1] (see also [45, Rem. 1] for an equivalent definition). Although the theorem holds for general interpolation models, it requires some geometric conditions on the interpolation set. The assumption of norm boundedness of the matrix Y^{-1} is equivalent to say that the set $\{y_2, \ldots, y_{n+1}\}$ is sufficiently linear independent. In [110], the authors propose a

QR-like variant algorithm that constructs points satisfying the hypotheses of Theorem 3.9, as proved in [110, Lem. 2.4] (see also [112, Algorithm 4.1]). We can thus formalize the convergence of our approach by dropping Assumption A3 but strengthening the rule for defining the set Y_{k+1}.

Theorem 3.10. *Consider problem* (1.2) *and suppose that assumptions A1 and A2 hold. Furthermore, assume that Algorithm 4.2 from [111] is used to create the set* Y_{k+1} *on line 31 of Algorithm 1. If* $G_{k+1} \supset Y_{k+1}$ *and the model* $\mathcal{M}^k$ *interpolates* φ *at points in* Y_k *for all iterations* $k = 0, 1, \ldots$*, then the convergence results of Theorem 3.8 hold.*

Proof. Algorithm 4.2 from [111] ensures that the points composing Y_{k+1} on line 31 of Algorithm 1 satisfy, under the stated interpolation condition, the assumptions of Theorem 3.9. Therefore, if Y_{k+1} is contained in G_{k+1} defining the model in (3.4), then Assumption A3 holds and Theorem 3.8 applies. □

The price to pay for having the strong results from Theorem 3.9 is the increase of the computational burden: the probability function φ needs to be evaluated at each one of the $(n + 1)$ points in Y_{k+1}, i.e., $(n + 1)$ integrals of dimension n need to be computed. However, not all the $(n+1)$ function evaluations need to be performed at every iteration of Algorithm 1: we may reuse/recycle some points in $Y_k \cap B(\hat{x}^{k+1}, \Delta_{k+1})$ to define Y_{k+1}. Such a strategy may render (employing Algorithm 4.2 from [111] for defining Y_{k+1}) attractive even for easier probability maximization problems whose probability distributions yield a formulæ for computing gradients: evaluating the gradient of φ when the latter follows a log-normal or Gaussian distribution requires solving m integrals of dimension $(m - 1)$. Hence, for those special probability distributions, the choice between our DFO algorithm and a derivative-based method will depend on the dimension n of decision variables and dimension m of the random vector. We highlight that for general probability distributions, a gradient formula may not be computationally implementable or practical due to its high complexity.

3.3 Efficiency condition for the subproblem's stationary point

In Section 3.2 we saw that the efficiency condition (3.6) that an approximate solution of the subproblem (3.5) should satisfy is an important tool to prove the global convergence of Algorithm 1. In this section we present an adaptation for maximization problems of the algorithm proposed in [85] that ensures the efficiency condition. From now on, $\langle \cdot, \cdot \rangle$ denotes the inner product in $\mathbb{R}^n$ and $\| \cdot \|$ the associated norm.

The algorithm below considers an inexact line search along the arc $d_k : \mathbb{R}_+ \to \mathbb{R}^n$ defined by

$$d_k(t) := \texttt{Proj}_X \left(\hat{x}^k + t \nabla \mathcal{M}^k(\hat{x}^k) \right) - \hat{x}^k. \tag{3.19}$$

Note that the stationarity measure of the subproblem, defined in (3.7), can be written as $\pi_k = \|d_k(1)\|$.

Algorithm 2. Computation of the new iterate

Input : $\hat{x}^k \in X$, $\Delta_k > 0$, $0 < \mu_1 < \mu_2 < 1$, $0 < \mu_3 \leq 1$, $0 < \nu_3 < \nu_1 \leq \nu_5$, $0 < \nu_2 \leq 1$ and $\nu_4 > 0$

1. Find t_k^A such that $\mathcal{M}^k\big(\hat{x}^k + d_k(t_k^A)\big) - \mathcal{M}^k(\hat{x}^k) \geq \mu_1 \left\langle \nabla \mathcal{M}^k(\hat{x}^k), d_k\left(t_k^A\right) \right\rangle$, with $\left\| d_k\left(t_k^A\right) \right\| \leq \nu_1 \Delta_k$ and $t_k^A \geq \nu_2 t_k^B$ or $t_k^A \geq \min\left\{ \frac{\nu_3 \Delta_k}{\|\nabla \mathcal{M}^k(\hat{x}^k)\|}, \nu_4 \right\}$, where t_k^B (if required) is some strictly positive number that satisfies $\mathcal{M}^k\big(\hat{x}^k + d_k(t_k^B)\big) - \mathcal{M}^k(\hat{x}^k) \leq \mu_2 \left\langle \nabla \mathcal{M}^k(\hat{x}^k), d_k\left(t_k^{\mathrm{B}}\right) \right\rangle$.
2. Choose s_k such that $\mathcal{M}^k(\hat{x}^k + s_k) - \mathcal{M}^k(\hat{x}^k) \geq \mu_3 \left(\mathcal{M}^k\big(\hat{x}^k + d_k(t_k^A)\big) - \mathcal{M}^k(\hat{x}^k) \right)$, $\|s_k\| \leq \nu_5 \Delta_k$, and $\hat{x}_k + s_k \in X$

Output: $x^{k+1} = \hat{x}_k + s_k$

Step 1 of Algorithm 2 is an inexact line search along the arc (3.19). By setting $t_k^A = t_k^B$, one obtains a variant of the Goldstein line search conditions [65]. Since $\mu_3 \in (0, 1]$, Step 2 requires that the model's increase issued by s_k is at least a fraction of the increasing given by $d_k(t_k^A)$.

Following the ideas of [85], we present the results ensuring that the approximate local solution $\hat{x}^k + s_k$ computed in Step 2 of Algorithm 2 satisfies the efficiency condition (3.6).

The first lemma shows that the conditions stated in the algorithm are compatible.

Lemma 3.11. *[85, Lem. 5] Consider the input parameters of Algorithm 2. Then, there exists a step s^k satisfying the conditions of Steps 1 and 2.*

To the next results, define the curvature of the model $\mathcal{M}^k$ at the point $\hat{x}^k \in X$ along the step $s \in \mathbb{R}^n$ as

$$\omega^k(s) := \frac{2}{\|s\|^2}\left(\mathcal{M}^k(\hat{x}^k + s) - \mathcal{M}^k(\hat{x}^k) - \langle \nabla \mathcal{M}^k(\hat{x}^k), s\rangle\right). \tag{3.20}$$

Assumption A2 and the compactness of X imply the following bounds.

Lemma 3.12. *[85, Lem. 6] Suppose that Assumptions A1 and A2 hold. Consider the model $\mathcal{M}^k$ and the stability center $\hat{x}^k \in X$ at the iteration k and $\kappa_{\mathcal{M}}$ the constant defined in* (3.9)*. For all $s \in \mathbb{R}^n$, satisfying $\hat{x}^k + s \in X$, there exists a finite constant $c_5 > 0$, independent of k, such that,*

$$i) \quad \|\nabla \mathcal{M}^k(\hat{x}^k)\| \leq c_5 \qquad \textit{and} \qquad ii) \quad |\omega^k(s)| \leq \kappa_{\mathcal{M}}. \tag{3.21}$$

Proof. By the triangle inequality and Lemma 3.2, there exists a constant $c_4 > 0$ such that

$$\|\nabla \mathcal{M}^k(\hat{x}^k)\| \leq \|\nabla \mathcal{M}^k(\hat{x}^k) - \nabla \varphi(\hat{x}^k)\| + \|\nabla \varphi(\hat{x}^k)\| \leq c_4 + \sup_{x\in X} \|\nabla \varphi(\hat{x}^k)\|.$$

As $\varphi \in \mathcal{C}^1$ and X is compact, we can define $c_5 = c_4 + \sup\limits_{x\in X} \|\nabla \varphi(\hat{x}^k)\|$, which proves (3.21) i). By (3.20), [64, Lem. 1.2.3] and Lemma 3.1, we have $|\omega^k(s)| = \dfrac{2}{\|s\|^2}\left|\mathcal{M}^k(\hat{x}^k + s) - \mathcal{M}^k(\hat{x}^k) - \langle \nabla \mathcal{M}^k(\hat{x}^k), s\rangle\right| \leq \kappa_{\mathcal{M}}$, proving (3.21) ii). □

We now state the result ensuring the efficiency condition (3.6) at iteration k.

Theorem 3.13. *Consider an iteration k of Algorithm 1. Suppose that $\hat{x}^k$ is not a stationary point of the subproblem* (3.5)*. Then, there exists a constant $c_1 > 0$, independent of k, such that the point x^{k+1}, computed by Algorithm 2, satisfies the efficiency condition $\mathcal{M}^k(x^{k+1}) - \mathcal{M}^k(\hat{x}^k) \geq c_1 \pi_k^2 \min\left\{\frac{\pi_k^2}{\beta}, \Delta_k, 1\right\}$.*

Proof. Consider $\omega_k = \omega^k\big(d_k(t_k^B)\big)$ if t_k^B is defined, and $\omega_k = 0$ otherwise. From [85, Thm.7], we have that $\omega_k \leq 0$ and $\mathcal{M}^k(x^{k+1}) - \mathcal{M}^k(\hat{x}^k) \geq c_1 \pi_k^2 \min\left\{\dfrac{\pi_k^2}{1-\omega_k}, \Delta_k\right\}$. Note that $1 - \omega_k \leq 1 + \kappa_{\mathcal{M}}$ from (3.21) ii). We complete the proof by denoting $\beta = 1 + \kappa_{\mathcal{M}}$ and noting that

$$\min\left\{\frac{\pi_k^2}{1-\omega_k}, \Delta_k\right\} \geq \min\left\{\frac{\pi_k^2}{\beta}, \Delta_k\right\} \geq \min\left\{\frac{\pi_k^2}{\beta}, \Delta_k, 1\right\}.$$

□

Chapter 4

Numerical experiments

In this chapter, we present numerical experiments for comparing the performance of Algorithm 1, deployed in two variants according to the rules discussed in Subsection 3.2.3, with other methods in the literature for solving two classes of problems.

First, we consider a class of continuous problems with three families of probability maximization problems, totalizing 90 instances. We assume that the random vector ξ follows two different multivariate elliptical distributions: the *Gaussian* and *Student's t*-distributions. The dimension of the decision variable varies from 3 to 566 and the dimension of random vector from 2 to 324. The variants of Algorithm 1 are benchmarked against several DFO solvers available in the literature.

Next, we assess the numerical performance of Algorithm 1 for solving a family of probability maximization problems with mixed-integer variables with dimension 36, being 12 integer variables. The dimension of the random vector is 12 and it follows a Gaussian distribution. In this case, our variants are compared with two derivative-based algorithms specialized in this class of problems.

All tests were performed on a Desktop Intel Core i7-7700K, CPU 4.20 GHz, 16GB RAM dual channel (3200 MHz), Windows 10 Pro 64 bits with codes in `Matlab` version R2018a.

4.1 Nonlinear continuous problems

In this section, we present numerical experiments for solving 90 instances from three families of probability maximization problems with the random vector ξ following two multivariate elliptical distributions: the *Gaussian* and the *Student's* t-distributions. First we describe the solvers, next the test problems and then the numerical results are discussed.

4.1.1 Solvers

Our two variants of Algorithm 1 are denoted by `TRCI` and `TRC`. They differ essentially by the choices of the set G_k in the problem (3.4b), the rule for updating the interpolation set Y_{k+1} on line 31, and by the set Λ at which the coefficients of the model $\mathcal{M}^k$ are defined.

- For `TRCI`, we set $\Lambda = \{\lambda \in \mathbb{R}^r : \|\lambda\|_\infty \leq 10^6\}$ and $G_k = Y_k$, $k = 1, 2 \ldots$, to define the model (3.4). The set Y_{k+1} on line 31 is constructed by the `Matlab` routine `AffPoints`[1] from [110], with default parameters. This routine constructs Y_{k+1} with $(n+1)$ linearly independent points in all iterations of Algorithm 1 as required by Theorem 3.9. As mentioned in Section 3.2.3, such strategy is expensive because φ needs to be evaluated many (but not more than $n+1$) times per iteration.

- `TRC` sets the model $\mathcal{M}^k$ as a convex combination of copulæ by taking Λ as the simplex in $\mathbb{R}^r$, $G_{k+1} = G_k \cup Y_{k+1} \cup \{x^{k+1}\}$, and the following simple rule for constructing Y_{k+1} on line 31: $Y_{k+1} = \emptyset$ if $\Delta_k \leq \alpha\pi_k$, and $Y_{k+1} = \{\hat{x}^k + \rho e_i\}$ otherwise, where $\rho = \min\{10^{-5}, \Delta_{k+1}\}$, e_i is the i^{th}-canonical direction and $i \in \{1, \ldots, n\}$ is randomly chosen, but avoiding the same one in two consecutive iterations. In this manner, only a single evaluation of φ is needed per iteration: either at the next iterate x^{k+1} if $\Delta_k \leq \alpha\pi_k$, or at $\hat{x}^k + \rho e_i$ otherwise. Convergence is guaranteed provided the conditions in Subsection 3.2.3 are fulfilled, yielding thus A3.

In both versions of Algorithm 1, the dictionary $\mathcal{D}_r$ is composed by 28 copulæ from five Archimedean families, as presented in Table 4.1. We tested two solvers for computing an

[1] Available at https://www.mcs.anl.gov/~wild/orbit/

Family	θ
Ali-Mikhail-Haq	-1, -0.2, 0, 0.5, 0.7, 0.99
Clayton	-1, -0.2, 0.2, 1, 3, 5, 7
Frank	-5, -1, 1, 5, 8
Gumbel-Hougaard	1, 2, 3, 7
Joe	1, 1.5, 2.5, 3, 4, 5

Table 4.1: Copulæ and their parameter θ of the dictionary $\mathcal{D}_r$.

approximate stationary point for subproblem (3.5), namely FilterSD[2], which is a Fortran 77 code interfaced by the `Matlab` OPTI Toolbox [21], and our implementation of the Frank-Wolfe algorithm [38] with Armijo line search and its parameters set as suggested in [115]. Depending on the family of test problems, one solver performed better than the other: subproblems were solved by Frank-Wolfe algorithm when ξ follows a Gaussian distribution, and by FilterSD when ξ follows a Student's t-distribution.

Both implementations of Algorithm 1 consider the same values for the trust-region parameters: $\eta = 0$, $\eta_1 = 0.2$, $\eta_2 = 0.6$, $\tau_1 = 0.5$, $\tau_2 = 2$, chosen from values suggested by [14, 108, 110] after some tuning. The other parameters were set as $\alpha = 10^8$, $\Delta_{max} = \min\{\max\{0.2\|x^0\|_\infty, 1\}, 20n\}$ and $\Delta_0 = 0.1\Delta_{max}$.

In order to validate our approaches, we compare their performance with six other derivative-free solvers:

- `TRL`: Derivative-free Trust-Region algorithm[3] [108] with Linear model and the same parameters of `TRC`.

- `TRQ`: Derivative-free Trust-Region algorithm [108] with Quadratic model and the same parameters of `TRC`.

- `COBYLA`: Constrained Optimization By Linear Approximation algorithm [68], available in the `Matlab` OPTI Toolbox [21].

[2]Available at `https://projects.coin-or.org/filterSD/`

[3]We are grateful to Dr. Adriano Verdério, from UTFPR Brazil, for providing us the codes of `TRL` and `TRQ`.

- `LINCOA`: LINearly Constrained Optimization Algorithm [69] available for `Matlab` in the `PDFO` package [79].

- `NOMAD`: Nonlinear Optimization with Mesh Adaptive Direct Search algorithm (MADS) [26], for inequality constrained problems, available in the `Matlab` OPTI Toolbox [21] with version 3.6.2.

- `PSwarm`[4]: Global optimization algorithm [107] for bound and linear inequality constrained problems, which combines pattern search and Particle Swarm strategies.

Algorithms `TRL`, `TRQ` and `LINCOA` are derivative-free algorithms that differ from our approaches by the construction of the model and the trust-region subproblem, since `TRL` considers linear polynomial interpolation and `TRQ` and `LINCOA` consider quadratic polynomial interpolation. One of the most relevant difference between the solvers `TRQ` and `LINCOA` is the number of points used in the interpolation set to construct the model. While `TRQ` considers $(n+1)(n+2)/2$ interpolation points, `LINCOA` considers a number between $n+2$ and $(n+1)(n+2)/2$. Another difference is that the trust-region subproblem of `LINCOA` is solved by the truncated conjugate gradient method, while `Gurobi`[5] is used for `TRQ`. We applied `TRL` and `TRQ` with the same trust-region parameters of our approaches because they presented better performance when compared with the default ones. The initial trust-region radius of `LINCOA` was set the same as `TRCI`. All linear and quadratic subproblems present in the `TRCI`, `TRC`, `TRL` and `TRQ` were solved by `Gurobi`.

In all solvers, the probability function φ was evaluated by the `Matlab` routines available in the *Truncated Normal and Student's t-distribution Toolbox*[6] and based on [8]: `mvnqmc` and `mvtqmc`, for the Gaussian and Student's t-distribution, respectively. These routines compute an estimator of the probability via Quasi Monte-Carlo simulation. In our tests, we set the number of simulations equal to 10 000.

The solvers `TRCI` and `TRC` stop when $\Delta_k \leq \texttt{tol}$, with $\texttt{tol} = 10^{-6}$, and one of the

[4]Available at `http://www.norg.uminho.pt/aivaz/pswarm/`

[5]Version 9.0.1, `www.gurobi.com`.

[6]Available at `https://www.mathworks.com/matlabcentral/fileexchange/53796-truncated-normal-and-student-s-t-distribution-toolbox`

following conditions hold:

$$\pi_k \leq \texttt{tol} \quad \text{or} \quad \Delta_k \leq \alpha\pi_k \quad \text{and} \quad |\varphi(x^k) - \varphi(x^{k-1})| \leq \texttt{tol} \quad \text{in 5 consecutive iterations.}$$

The other solvers were applied with default stopping criteria, except by the tolerance in the criterion $\Delta_k \leq \texttt{tol}$, considered by `TRL` and `TRQ`, which was the same as in `TRCI` and `TRC`. Furthermore, a CPU time limit of one hour was given to all solvers, and the maximum number of objective function evaluations was set to $100(n+1)$.

4.1.2 Test problems and numerical experiments

We consider 90 instances in three different sets of probability maximization problems, originally formulated as CCPs (1.3), where f is a linear function, $\tilde{X}$ is a polytope, φ is given by (1.2) with g a linear mapping. Two different distributions for the random vector ξ are examined: a Gaussian one with given positive definite covariance matrix Cov_G, and a Student's t-distribution with $\nu = 4$ degrees of freedom and covariance matrix $Cov_T = \frac{\nu}{\nu-2} Cov_G = 2Cov_G$. As in [74], we reformulated the problems as PMPs by defining $X := \{x \in \tilde{X} : f(x) \leq T\}$, where $T = \tau f(x^0)$, with τ a given target and $x^0 \in \tilde{X}$ an initial point. We consider six uniformly spaced values for the parameter τ as described in Table 4.2. These values start at 1, corresponding to the lowest probability, and go to $\bar{\tau}$ related to an optimal probability value close to 1, obtained by solving (1.2) with a Gaussian distribution. The value $\bar{\tau}$ can be greater or smaller than 1, depending on the problem. The initial point x^0 was set as a solution of the simpler individual chance-constrained problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & \mathbb{P}\big[\xi_i \leq g_i(x)\big] \geq 0.95, \quad i = 1, \ldots, m \\ & x \in \tilde{X}. \end{array} \tag{4.1}$$

With the help of p-quantiles, the individual probability constraints can be written as linear ones: $\mathbb{P}^{-1}[p] \leq g_i(x)$, $i = 1, \ldots, m$. Thus, (4.1) becomes a linear problem because g, f,

and the constraints that define $\tilde{X}$ are linear functions. We highlight the small change in notation that significantly impacts the problem's nature: the difficult joint-probability is denoted by $\mathbb{P}\big[\xi_i \leq g_i(x),\ i = 1, \ldots, m\big]$, whereas the much simpler individual probabilities are $\mathbb{P}\big[\xi_i \leq g_i(x)\big],\ i = 1, \ldots, m$.

The 90 instances of test problems are summarized in Table 4.2. The first and second columns indicate the type and name of the problems; the third and fourth give the dimensions of the decision and random variables, respectively; the fifth column shows the type of the probability distribution of ξ; the sixth provides the average (over 1000 points) of CPU time $\bar{t}$ (in seconds) required to evaluate the probability function φ, i.e., the oracle CPU time; the seventh discriminates the values of the parameter τ used to define X above; the eighth summarizes the number of problem instances; and the last column indicates the DFO solvers under comparison for every set of problems. As `NOMAD` and `PSwarm` handle only inequality constrained problems, they have not been considered for solving the second set of problems that involve equality constraints. Furthermore, `TRC` was the only DFO algorithm capable to solve the third set of problems within the time limit of one hour.

Type	Problem	n	m	Distribution	$\bar{t}$	τ	# inst.	DFO solvers
Academic: ineq. constraints	Cash matching	3	15	Gaussian	0.082	0.900 0.920 0.940 0.960 0.980 1.000	12	`TRC`, `TRCI` `TRL`, `TRQ` `COBYLA` `LINCOA` `NOMAD` `PSwarm`
				Student	0.096			
	Transport	8	4	Gaussian	0.024	1.000 1.040 1.080 1.120 1.160 1.200	24	
				Student	0.030			
		32	8	Gaussian	0.047			
				Student	0.054			
Academic: ineq. and eq. constraints	PlanToy	8	2	Gaussian	0.001	1.000 1.014 1.028 1.042 1.056 1.070	36	`TRC`, `TRCI` `TRL`, `TRQ` `COBYLA` `LINCOA`
				Student	0.018			
Industrial: ineq. and eq. constraints	Reservoir	566	96	Gaussian	0.400	0.995 0.996 0.997 0.998 0.999 1.000	18	`TRC`
		566	192		1.190			
		566	324		4.722			

Table 4.2: Information about the test problems. Notation $\bar{t}$ stands for the estimated CPU time in seconds required to evaluate the objective function (oracle call).

Academic problems with inequality constraints

We consider two families of academic problems, each one defined with Gaussian and Student's t-distribution.

Cash Matching problem. This is a PMP variant of the well known chance-constrained problem presented in [48]. The goal is to make a portfolio, with a certain amount of cash, of n types of bonds on behalf of a pension fund that maximizes the probability of covering certain payments over the coming m time periods while satisfying that the sum of the bond yields, at the end of the period, reaches a minimal target. The decision vector $x \in \mathbb{R}^n$ ($n = 3$) corresponds to the amount of each type of bond to be bought and the random vector $\xi \in \mathbb{R}^m$ ($m = 15$) represents the payments of the time periods. The problem's data can be found in [48].

Probabilistic Transportation problem. This is a PMP version of the stochastic transportation problem from [59]. The goal is to maximize the probability of satisfying a random demand of products shipped from a set S of suppliers to a set C of customers, while ensuring the supply capacity is respected and the shipment costs are not higher than a given budget (target). The decision variable $x \in \mathbb{R}^n$ ($n = |C|\,|S|$, where $|Z|$ is the cardinality of the set Z), is the amount of products shipped from the suppliers to the customers and the random vector $\xi \in \mathbb{R}^m$ represents the demands. We considered two different pairs of values for the number of suppliers $|S|$ and customers $|C|$, i.e., $(|C|, |S|) \in \{(4, 2), (10, 6)\}$. Data were randomly generated according to [59].

Table 4.3 reports the results obtained by all eight derivative-free algorithms considered for solving the 36 instances of these two problems. The three first columns refer to the problem data, and the others refer to the computed functional value, number of function evaluations, and CPU time in seconds for each solver. Using the criterion proposed in [9], as the image of φ lies in $[0, 1]$, we say that an algorithm solves a problem if it finds a point

$\bar{x} \in X$ such that

$$\frac{|\varphi_{\max} - \varphi(\bar{x})|}{\max\{1, |\varphi(\bar{x})|, |\varphi_{\max}|\}} = |\varphi_{\max} - \varphi(\bar{x})| \leq 10^{-2}, \tag{4.2}$$

where $\varphi_{\max}$ is the largest function value computed by the solvers under comparison. The symbol † next to the function value in the table indicates that the algorithm did not solve the problem according to this criterion.

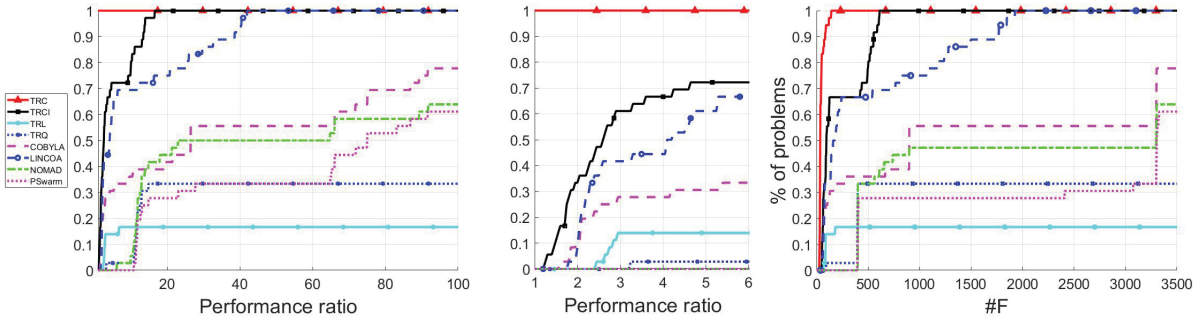
We also present data and performance profiles [27, 61] for the number of function evaluations $\#F$ and for CPU time (in seconds). As suggested in [4], we say that two algorithms tie in respect to CPU time, if the difference of time spent by them is less than 5% of the time spent by the fastest algorithm to solve a given problem. Figures 4.1 and 4.2 show the profiles with respect to the number of function evaluations and CPU time, respectively. Figures 4.1a and 4.2a present performance profiles with a zoomed view, while Figures 4.1b and 4.2b show data profiles.

Figure 4.1 shows that the most robust and efficient algorithm in terms of function evaluations is `TRC`, solving 100% of the problems with the minimal amount of oracle calls: every instance was solved by `TRC` with at most 144 function evaluations. Algorithms `TRCI` and `LINCOA` also solved all problems, but using 16.6 and 42 times the number of function evaluations required by `TRC`, respectively, and no more than 613 and 1927 evaluations per problem instance. On the other hand, `TRL`, `TRQ`, `COBYLA`, `NOMAD` and `PSwarm` solved 16.7%, 33.3%, 77.8%, 63.9% and 61.1% of the problems, respectively. From Figure 4.1b we see that when the solvers perform at most 500 objective function evaluations, `TRC`, `TRCI`, `TRL`, `TRQ`, `COBYLA`, `LINCOA`, `NOMAD` and `PSwarm` solved 100%, 77.8%, 16.7%, 33.0%, 36.1%, 66,7%, 33.3% and 27.8% of the problems, respectively.

Concerning CPU time, Figure 4.2a indicates that `TRC` remains the most robust and efficient algorithm, solving 69.4% of all instances with the lowest CPU time. Solvers `COBYLA` and `LINCOA` solved 13.9% and 25% of the problems with the lowest time, respectively. From Figure 4.2b we see that when it is allowed to spend at most 100 seconds, `TRC` and `TRCI` solve all problems, while `TRL`, `TRQ`, `COBYLA`, `LINCOA`, `NOMAD` and `PSwarm` solve 16.7%, 33.3%, 55.6%, 97.2%, 47.2% and 27.8%, respectively.

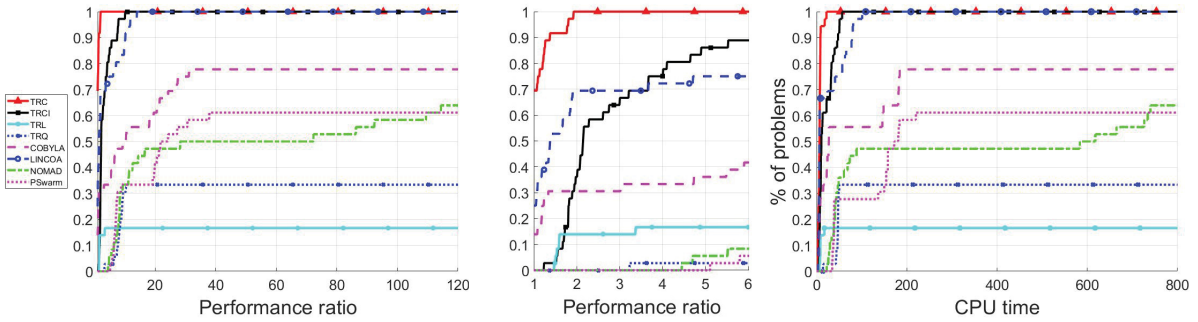
Table 4.3: Academic problems with inequality constraints: computed function value, number of function evaluations, and CPU time spent by all eight DFO solvers.

	Problem data		$\varphi(\bar{x}^k)$								#F								CPU time							
	Problem	τ	TRC	TRCI	TRL	TRQ	COBYLA	LINCOA	NOMAD	PSwarm	TRC	TRCI	TRL	TRQ	COBYLA	LINCOA	NOMAD	PSwarm	TRC	TRCI	TRL	TRQ	COBYLA	LINCOA	NOMAD	PSwarm
Gaussian distribution	Cash matching	0.900	0.992	0.992	0.936†	0.992	0.992	0.992	0.992	0.989	37	63	272	400	108	80	402	403	6.87	22.09	23.34	42.97	9.16	6.95	37.95	35.13
		0.920	0.986	0.986	0.936†	0.986	0.986	0.986	0.986	0.983	35	71	272	400	88	76	402	401	5.99	22.05	23.04	43.95	7.43	6.33	37.98	34.80
		0.940	0.976	0.976	0.936†	0.976	0.976	0.976	0.976	0.968	34	58	272	400	70	80	402	400	5.13	9.26	23.63	46.80	6.14	6.61	73.44	35.55
		0.960	0.962	0.962	0.936†	0.962	0.962	0.962	0.962	0.961	36	54	272	400	57	75	402	408	5.70	7.96	23.21	48.86	5.22	6.19	40.89	37.00
		0.980	0.941	0.941	0.941	0.941	0.941	0.941	0.941	0.923 †	31	74	87	400	166	80	402	410	4.83	9.38	7.67	50.21	14.62	6.60	60.68	37.29
		1.000	0.910	0.910	0.901	0.906	0.910	0.910	0.910	0.888 †	29	64	71	94	277	73	402	403	4.38	9.41	6.46	14.16	24.06	6.04	40.46	36.77
	Transport n = 8 m = 4	1.000	0.926	0.926	0.856†	0.905†	0.926	0.926	0.919	0.844 †	31	102	47	900	130	198	658	909	3.10	6.65	1.18	68.97	3.15	5.68	26.00	22.09
		1.040	0.952	0.952	0.886†	0.920†	0.932†	0.952	0.918 †	0.844 †	67	121	56	900	901	136	485	908	6.21	6.99	1.47	43.83	21.62	3.23	18.14	22.40
		1.080	0.970	0.970	0.882†	0.959†	0.961	0.970	0.958 †	0.844 †	42	96	47	269	901	222	902	910	3.83	6.52	1.20	29.08	22.62	5.32	38.34	22.59
		1.120	0.982	0.982	0.908†	0.948†	0.969†	0.982	0.981	0.844 †	74	90	89	900	901	242	740	910	7.05	7.25	2.36	46.31	21.79	5.87	27.63	22.24
		1.160	0.989	0.989	0.908†	0.969†	0.965†	0.989	0.989	0.844 †	92	116	89	900	901	208	563	910	9.09	9.10	2.28	49.15	22.59	5.03	22.42	22.22
		1.200	0.994	0.994	0.908†	0.966†	0.987	0.994	0.988	0.844 †	85	123	89	125	901	192	902	910	8.32	9.47	2.33	12.53	22.24	4.69	33.37	22.11
	Transport n = 32 m = 8	1.000	0.766	0.765	0.704†	0.736†	0.762	0.765	0.763	0.767	50	504	67	586	3301	1280	3302	3301	8.07	29.60	3.26	35.85	146.12	56.89	583.40	154.39
		1.080	0.845	0.845	0.692†	0.816†	0.845	0.843	0.833 †	0.843	44	440	54	583	3301	1770	3302	3300	7.65	31.41	2.67	33.79	148.41	77.44	645.02	149.90
		1.160	0.902	0.901	0.699†	0.855†	0.900	0.901	0.876 †	0.901	44	590	64	583	3301	1848	3302	3301	7.09	33.02	2.99	35.14	145.79	81.10	650.92	154.35
		1.240	0.941	0.941	0.777†	0.885†	0.840†	0.941	0.926 †	0.941	120	555	55	583	3301	544	3302	3327	18.01	32.47	2.53	42.89	147.78	24.28	622.77	157.91
		1.320	0.966	0.966	0.835†	0.899†	0.919†	0.966	0.958	0.963	51	529	55	583	3301	833	3302	3309	7.95	31.89	2.70	37.99	153.06	37.53	734.18	158.13
		1.400	0.981	0.981	0.872†	0.922†	0.946†	0.981	0.974	0.980	144	607	55	583	3301	1771	3302	3318	21.84	39.82	2.75	36.22	156.83	80.26	617.88	157.82
Student's T-distribution	Cash matching	0.900	0.936	0.936	0.921†	0.936	0.936	0.936	0.936	0.935	34	52	262	400	67	70	402	402	8.47	10.35	24.98	47.25	6.17	6.43	68.65	38.84
		0.920	0.931	0.932	0.921†	0.932	0.932	0.932	0.932	0.932	35	63	262	400	64	70	402	403	7.65	12.87	25.24	47.64	6.03	6.45	48.87	38.50
		0.940	0.925	0.925	0.919	0.925	0.925	0.925	0.925	0.924	34	53	90	400	71	61	402	400	7.04	10.68	8.65	47.65	6.70	5.64	47.27	38.13
		0.960	0.916	0.917	0.913	0.917	0.917	0.917	0.917	0.914	32	61	88	400	66	64	402	405	6.62	12.09	8.77	47.81	6.32	5.84	43.84	38.53
		0.980	0.906	0.907	0.907	0.907	0.907	0.907	0.907	0.900	27	65	182	400	64	55	402	401	5.52	13.38	17.26	46.27	6.01	5.12	42.36	38.45
		1.000	0.895	0.896	0.895	0.896	0.896	0.896	0.896	0.893	31	54	90	400	56	63	402	404	6.54	10.54	8.54	47.18	5.35	5.78	88.41	38.58
	Transport n = 8 m = 4	1.000	0.919	0.920	0.875†	0.906†	0.918	0.920	0.890 †	0.353 †	34	97	48	900	901	138	569	900	4.12	6.42	1.47	63.15	26.86	4.52	25.93	29.34
		1.080	0.932	0.933	0.894†	0.910†	0.933	0.933	0.911 †	0.850 †	35	125	48	900	667	142	790	910	2.69	8.09	1.41	49.89	20.19	4.32	36.16	27.64
		1.160	0.942	0.943	0.906†	0.912†	0.943	0.943	0.934	0.850 †	34	96	47	900	901	143	611	910	2.57	7.07	1.41	49.83	27.12	4.30	29.22	27.45
		1.240	0.951	0.951	0.907†	0.912†	0.947	0.951	0.930 †	0.850 †	34	89	46	900	901	179	558	910	2.93	5.07	1.37	50.00	27.14	5.44	25.17	27.86
		1.320	0.958	0.958	0.907†	0.912†	0.947†	0.958	0.944 †	0.850 †	34	88	46	900	901	158	678	910	2.66	5.54	1.37	52.11	26.74	5.00	30.91	27.57
		1.400	0.963	0.963	0.907†	0.912†	0.959	0.963	0.947 †	0.850 †	34	91	46	900	901	161	584	910	2.53	5.80	1.32	52.39	26.34	4.82	27.43	27.59
	Transport n = 32 m = 8	1.000	0.786	0.787	0.745†	0.750†	0.787	0.789	0.741 †	0.789	46	425	55	593	3301	1192	1616	3303	7.47	36.63	3.02	44.47	181.46	62.99	251.89	180.97
		1.080	0.823	0.824	0.000†	0.774†	0.825	0.826	0.819	0.825	50	554	53	583	3301	1927	3302	3303	8.58	47.52	2.96	38.50	184.22	101.17	739.97	182.10
		1.160	0.854	0.854	0.000†	0.805†	0.787†	0.856	0.845 †	0.856	37	613	46	584	3301	1498	3302	3074	5.61	56.64	2.57	36.51	182.95	78.40	599.89	171.88
		1.240	0.878	0.879	0.820†	0.824†	0.876	0.880	0.876	0.880	37	517	55	584	3301	767	3302	2409	6.64	50.82	2.97	45.93	182.37	40.85	726.48	136.38
		1.320	0.899	0.899	0.000†	0.839†	0.900	0.898	0.885 †	0.900	38	497	36	584	3301	1241	3302	3301	6.75	50.42	2.05	44.43	183.04	65.78	548.70	183.75
		1.400	0.915	0.915	0.000†	0.846†	0.915	0.916	0.910	0.916	36	494	38	584	3301	1067	3302	3300	5.81	45.62	2.09	39.55	180.84	56.81	664.97	220.11



(a) Performance profile (b) Data profile

Figure 4.1: Performance profile with a zoomed view (a) and data profile (b) with respect to the number of function evaluations of all eight DFO algorithms for solving the set of academic problems with inequality constraints.



(a) Performance profile (b) Data profile

Figure 4.2: Performance profile with a zoomed view (a) and data profile (b) with respect to CPU time of all eight DFO algorithms for solving the set of academic problems with inequality constraints.

Academic problems with inequality and equality constraints

PlanToy. This is a family of problems that consists of a two-month planning period of two fictitious oil refineries as described in [22, Sec.6.2.1]. The goal is to find a plan for processing, storing and importing two types of oil to maximize the probability of meeting the random demand ξ of fuels. More specifically, the objective is to maximize the probability of satisfying the random second-month demand while fulfilling deterministic constraints

such as storage capacity, first-month demand, and monetary budget. In this example, the decision variable $x \in \mathbb{R}^n$ ($n = 8$) represents the operation planning of the refineries and the random vector $\xi \in \mathbb{R}^m$ ($m = 2$) corresponds to second-month demand of fuels. The vector $\xi = (\xi_1, \xi_2)$ has mean $\mathbb{E}[\xi] = (193, 178)$ and, in the Gaussian setting, the covariance matrix is given by

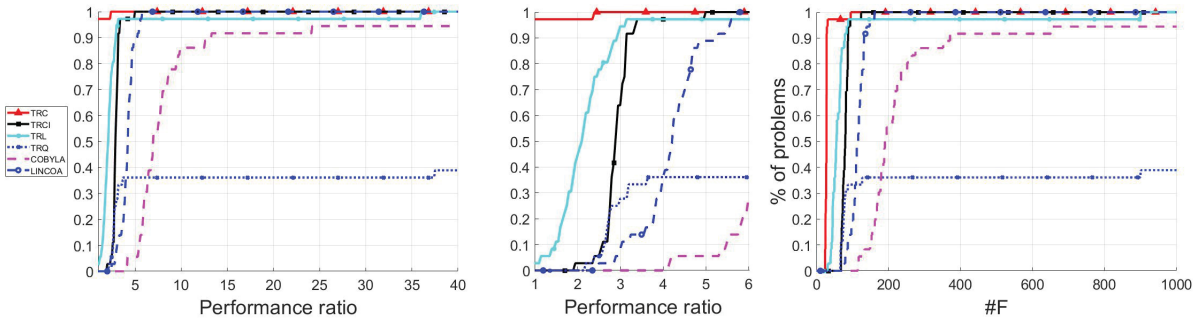
$$Cov_G = \begin{pmatrix} 9 & \texttt{Cov}\,(\xi_1, \xi_2) \\ \texttt{Cov}\,(\xi_1, \xi_2) & 10.24 \end{pmatrix}, \quad \text{with} \quad \texttt{Cov}(\xi_1, \xi_2) \in \{-4.8, 0, 4.8\}. \tag{4.3}$$

As mentioned above, in the Student t-distribution setting, the convariance matrix is $2Cov_G$. Table 4.4 reports on the results of six (out of eight) derivative-free algorithms for solving the 36 instances of PlanToy. Solvers `NOMAD` and `PSwarm` were removed from the comparison because they are not applicable to problems with equality constraints according to the user guides.

Table 4.4: Academic problems with inequality and equality constraints (PlanToy): computed function value, number of function evaluations, and CPU time spent by six DFO solvers.

	Problem data		$\varphi(\hat{x}^k)$						#F						CPU time					
	τ	$\mathtt{Cov}(\xi_1,\xi_2)$	TRC	TRCI	TRL	TRQ	COBYLA	LINCOA	TRC	TRCI	TRL	TRQ	COBYLA	LINCOA	TRC	TRCI	TRL	TRQ	COBYLA	LINCOA
Gaussian distribution	1.000	-4.8	0.935	0.935	0.935	0.934	0.935	0.935	25	68	900	80	213	110	0.64	0.45	1.37	10.88	1.53	2.02
	1.014	-4.8	0.954	0.954	0.954	0.952	0.954	0.954	25	68	54	75	211	121	0.36	0.41	0.15	5.88	1.03	0.13
	1.028	-4.8	0.968	0.968	0.968	0.958†	0.968	0.968	25	69	45	76	186	139	0.32	0.41	0.12	3.24	0.82	0.13
	1.042	-4.8	0.978	0.978	0.978	0.970	0.978	0.978	24	73	67	87	164	97	0.32	0.53	0.21	7.20	0.68	0.09
	1.056	-4.8	0.986	0.986	0.986	0.830†	0.986	0.986	25	124	52	142	190	116	0.33	0.76	0.14	30.83	0.81	0.16
	1.700	-4.8	0.991	0.991	0.991	0.977†	0.991	0.991	27	74	50	71	215	102	0.37	0.44	0.14	3.42	0.88	0.09
	1.000	0	0.935	0.935	0.935	0.932	0.895†	0.935	24	68	57	67	394	132	0.32	0.38	0.17	3.20	1.66	0.14
	1.014	0	0.954	0.954	0.954	0.953	0.954	0.954	26	72	55	67	252	110	0.34	0.40	0.16	3.54	0.90	0.09
	1.028	0	0.968	0.968	0.968	0.926†	0.968	0.968	27	70	54	73	188	112	0.35	0.41	0.13	4.26	0.77	0.10
	1.042	0	0.979	0.979	0.979	0.932†	0.979	0.979	25	70	45	105	195	106	0.31	0.50	0.12	23.04	0.84	0.12
	1.056	0	0.986	0.986	0.986	0.981	0.986	0.986	24	80	70	900	133	91	0.33	0.56	0.18	8.83	0.54	0.13
	1.700	0	0.991	0.991	0.991	0.980†	0.991	0.991	26	74	50	72	155	107	0.34	0.46	0.15	3.39	0.64	0.13
	1.000	4.8	0.937	0.937	0.937	0.931	0.937	0.937	31	73	67	68	217	76	0.42	0.43	0.18	3.52	1.13	0.09
	1.014	4.8	0.955	0.955	0.955	0.954	0.955	0.955	28	71	31	66	224	83	0.44	0.46	0.08	3.39	0.94	0.08
	1.028	4.8	0.969	0.969	0.969	0.865†	0.872†	0.969	27	75	69	70	240	97	0.40	0.48	0.19	2.70	0.99	0.08
	1.042	4.8	0.979	0.979	0.979	0.907†	0.979	0.979	27	82	46	900	235	103	0.37	0.58	0.13	26.21	0.95	0.10
	1.056	4.8	0.986	0.986	0.986	0.981	0.986	0.986	27	79	67	73	652	87	0.36	0.63	0.18	4.70	2.23	0.08
	1.700	4.8	0.991	0.991	0.991	0.988	0.991	0.991	95	78	40	127	235	122	5.01	0.67	0.10	28.98	0.81	0.10
Student's T-distribution	1.000	-4.8	0.931	0.931	0.931	0.920†	0.931	0.931	28	81	64	74	162	107	1.44	2.35	1.21	5.62	3.63	2.07
	1.014	-4.8	0.939	0.939	0.939	0.870†	0.939	0.939	28	94	88	82	371	112	1.12	2.72	1.66	14.97	7.03	1.98
	1.028	-4.8	0.947	0.947	0.947	0.921†	0.946	0.947	28	82	65	86	179	118	1.15	2.20	1.29	7.57	3.31	2.08
	1.042	-4.8	0.953	0.953	0.953	0.943†	0.953	0.953	28	82	66	85	117	161	1.12	2.23	1.25	7.93	2.07	2.81
	1.056	-4.8	0.959	0.959	0.959	0.000†	0.959	0.959	28	87	79	79	178	117	1.11	2.44	1.52	5.53	3.31	2.11
	1.700	-4.8	0.964	0.964	0.964	0.942†	0.964	0.964	28	86	53	79	153	131	1.11	2.33	1.07	9.25	2.72	2.30
	1.000	0	0.932	0.932	0.932	0.929	0.932	0.932	28	88	44	75	217	112	1.13	2.47	0.86	6.71	4.06	1.96
	1.014	0	0.941	0.941	0.941	0.936	0.941	0.941	28	88	44	74	195	80	1.13	2.45	0.84	8.38	3.54	1.39
	1.028	0	0.948	0.948	0.948	0.921†	0.948	0.948	28	80	47	77	173	148	1.12	2.19	0.89	7.58	3.11	2.63
	1.042	0	0.954	0.954	0.954	0.930†	0.954	0.954	28	81	77	90	275	120	1.13	2.30	1.62	14.87	4.96	2.07
	1.056	0	0.960	0.960	0.960	0.952	0.960	0.960	29	85	56	78	169	132	1.24	2.43	1.09	9.60	3.24	2.28
	1.700	0	0.965	0.965	0.965	0.000†	0.965	0.965	28	87	64	103	115	127	1.11	2.66	1.22	11.62	2.14	2.18
	1.000	4.8	0.936	0.936	0.936	0.004†	0.936	0.936	28	87	57	127	188	110	1.19	2.46	1.15	39.29	3.81	1.91
	1.014	4.8	0.944	0.944	0.944	0.932†	0.944	0.944	28	87	42	110	180	131	1.14	2.40	0.83	27.00	3.40	2.26
	1.028	4.8	0.951	0.951	0.951	0.941†	0.951	0.951	28	76	84	83	149	125	1.15	2.13	1.71	6.41	2.89	2.15
	1.042	4.8	0.957	0.957	0.957	0.932†	0.957	0.957	28	85	60	78	265	118	1.16	2.46	1.14	6.37	4.94	2.04
	1.056	4.8	0.962	0.962	0.962	0.955	0.962	0.962	28	78	39	77	352	131	1.19	2.23	0.76	7.76	6.61	2.26
	1.700	4.8	0.966	0.966	0.966	0.943†	0.966	0.966	28	81	45	91	168	123	1.15	2.25	0.84	12.13	3.02	2.10

Figure 4.3a shows that the most robust and efficient algorithm in terms of function evaluation is `TRC`, solving 97.2% of the problems with the minimal amount of oracle calls, while `TRL` is the most efficient algorithm only in 2.8% of the problems. The algorithms `TRC`, `TRCI`, `TRL` and `LINCOA` solved all problems while `TRQ` and `COBYLA` solved 38.9% and 94.4%, using at most 2.4, 5.0, 36.0, 5.7, 38.9 and 24.2 times the number of function evaluations required by the best algorithm, respectively. From Figure 4.3b we see that when the solvers perform at most 200 objective function evaluations, `TRC`, `TRCI` and `LINCOA` solved all problems, while `TRL`, `TRQ` and `COBYLA` solved 97.2%, 36.1% and 55.6%, respectively.

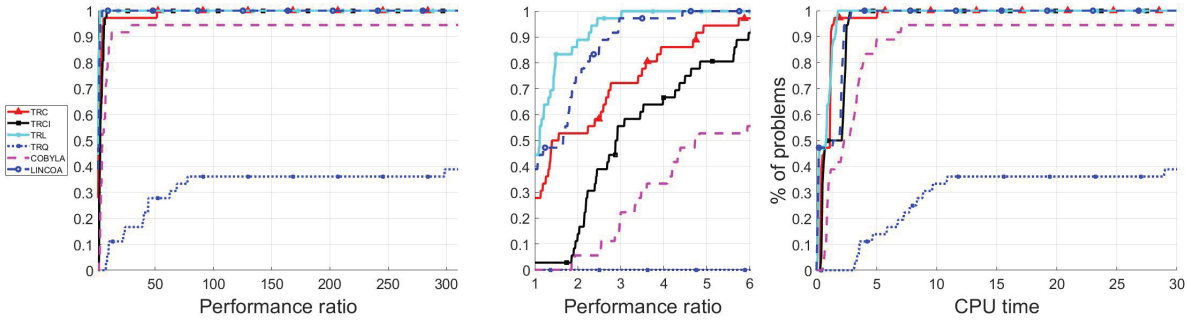


(a) Performance profile (b) Data profile

Figure 4.3: Performance profile with a zoomed view (a) and data profile (b) with respect to the number of function evaluations of the six DFO algorithms for solving the set of academic problems with inequality and equality constraints.

Figure 4.4a indicates that `TRL` was the most efficient solver, solving 44.4% of the instances with the best CPU time, while `LINCOA`, `TRC` and `TRCI` solved 38.9%, 27.8% and 2.8%, respectively. From Figure 4.4b we see that when it is allowed to spend at most 6 seconds, `TRC`, `TRCI`, `TRL` and `LINCOA` solve all problems, while `TRQ` and `COBYLA`, solve 16.7%, 88.9%, respectively.

In this set of problems, `TRL` was the most efficient solver in terms of CPU time, but not in terms of the number of function evaluations. Since the PlanToy family has a random vector of dimension 2, the cost for evaluating the probability function, in relation to CPU time, is not as impactful as solving the trust-region subproblem: recall that `TRL` solves a linear program per iteration using `Gurobi`, while `TRC` and `TRCI` solve a nonlinear program.



(a) Performance profile (b) Data profile

Figure 4.4: Performance profile with a zoomed view (a) and data profile (b) with respect to CPU time of the six DFO algorithms for solving the set of academic problems with inequality and equality constraints.

When the dimension of the random vector is higher, evaluating φ becomes the solvers' bottleneck, as evidenced in the next results (see also the sixth column in Table 4.2).

Industrial problems

Cascaded-Reservoir Management problems. This is a family of energy planning problems with a real-life configuration of a French hydro valley, described in [95, 105]. The objective is to maximize the probability that reservoirs' volumes remain within bounds and the profit yielded by power generation decisions reaches a minimal target. The decision variable $x \in \mathbb{R}^n$ $(n = 566)$ represents the operation planning of power units while the vector $\bar{\xi}$ corresponds to random water inflows. Since the original data contains a bilateral inequality under the probability function, i.e., $\mathbb{P}[Ax \leq \bar{\xi} \leq Bx]$, Sklar's theorem is not directly applicable. For purposes of Algorithm 1, we adopt the reformulation given by (2.6) and (2.7), fitting thus the structure in (1.2) by considering a random vector $\xi \in \mathbb{R}^m$ with twice as many random data, i.e, $\xi = [-\bar{\xi}, \bar{\xi}]$. Three instances for this vector were considered with dimension $m \in \{96, 192, 324\}$, according to the three *Max-P* models from [105].

We consider 18 instances of the cascaded-reservoir management problem. With exception of `TRC`, all others derivative-free algorithms failed to solve these instances within the

time limit of one hour (in practice, variants of these problems should be solved at every thirty minutes to rend on-time power generation dispatches). We point out that the dimension $n = 566$ of the decision variable is remarkable for DFO algorithms. For instance, this dimension rendered `TRQ` impracticable because $161028 = (566 + 1)(566 + 2)/2$ function evaluations would be necessary only at the first iteration to construct the underlying quadratic model, resulting in approximated 18 hours of CPU time when the dimension of the random vector is $m = 96$. Each function evaluation takes approximately 0.4 seconds, as displayed in Table 4.2. The situation is even more complicated when $m = 324$: evaluating the function at a single point x takes around 4.7 seconds. Although this difficulty and the large dimension of the decision vector, our variant `TRC` of Algorithm 1 was able to solve each one of the 18 instances in at most 13.3 minutes, as indicated in Table 4.5. This is thanks to the fact that `TRC` requires only a function evaluation per iteration. The variant `TRCI` that needs at most n extra points to build the model, would spend, for the cases $m = 96$, $m = 192$, and $m = 324$, up to 3.7, 11.2, and 44.5 minutes, respectively, per iteration only to evaluate the objective function at these points.

In order to provide another solver to benchmark `TRC`, we exploited the fact that there is an implementable formula for computing the gradients of φ when ξ follows a Gaussian distribution and, moreover, $-\log(\varphi)$ is convex. As a result, we can reformulate (1.2) as a typical nonlinear, differentiable, and convex optimization problem. We tested several NLP (derivative-based) solvers available in the literature, and report results only for the most successful one in our experiments: the Level Bundle method, denoted by `LB`, with default stopping criteria and parameters as described in [55] (see also [96] for experiments on the same class of problems). Table 4.5 presents the results of `TRC` and `LB` on the considered 18 instances of the problem. The two first columns report on the problem's data. The other columns provide information on the computed function value, number of function and gradient evaluations, iterations and CPU time. As we can see from the table, solver `LB` stopped by reaching the time limit of one hour in the instances with $m = 192$ and $m = 324$. This highlights how expensive it is to compute derivatives of the probability function with high-dimensional random vectors: roughly, a first-order oracle is m times

more time-consuming than the zero-order oracle.

Table 4.5: Industrial problems: computed function value, number of function and gradient evaluations, and CPU time (sec) spent by `TRC` and `LB`.

Problem data		$\varphi(\hat{x}^k)$		#F		#G		CPU time	
τ	m	`TRC`	`LB`	`TRC`	`LB`	`TRC`	`LB`	`TRC`	`LB`
0.995	96	0.996	0.996	51	47	0	47	35.6	1163.4
0.996	96	0.992	0.992	51	54	0	54	33.0	1362.0
0.997	96	0.985	0.985	48	34	0	34	31.3	861.4
0.998	96	0.972	0.972	46	36	0	36	28.4	911.6
0.999	96	0.949	0.950	48	34	0	34	34.6	860.9
1.000	96	0.912	0.913	43	23	0	23	29.2	556.4
0.995	192	0.968	0.927†	61	21	0	21	281.8	3758.2
0.996	192	0.952	0.920†	54	21	0	21	229.5	3735.6
0.997	192	0.930	0.932	50	21	0	21	228.8	3733.1
0.998	192	0.900	0.900	46	21	0	21	172.1	3708.6
0.999	192	0.858†	0.869	47	21	0	21	161.8	3729.5
1.000	192	0.804†	0.818	44	21	0	21	158.8	3758.6
0.995	324	0.873	0.738†	60	5	0	5	798.1	4250.9
0.996	324	0.851	0.739†	53	5	0	5	639.2	4258.3
0.997	324	0.823	0.725†	63	5	0	5	705.4	4231.0
0.998	324	0.787	0.724†	48	5	0	5	519.0	4074.9
0.999	324	0.746	0.727†	43	5	0	5	417.9	4206.3
1.000	324	0.706	0.684†	48	5	0	5	568.3	4276.8

4.2 Mixed-Integer Nonlinear Programming problems

The promising numerical results of our approaches `TRCI` and `TRC` to the nonlinear continuous problems piqued our curiosity to investigate their numerical performance to Mixed-Integer Nonlinear Programming Programming (MINLP) problems. Analogously to the discussion of Chapter 1, for a given convex set $\tilde{X} \subset R^{n_x}$ and a set containing only integer variables $\tilde{Y} \subset \mathbb{Z}^{n_y}$, both compact sets, the general probability maximization MINLP problem can be represented as

$$\begin{aligned} \text{maximize} \quad & \mathbb{P}\left[\xi \leq g(x,y)\right] \\ \text{subject to} \quad & (x,y) \in X \times Y, \end{aligned} \tag{4.4}$$

where $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \mapsto \mathbb{R}^m$ and $X \times Y = \{(x, y) \in \tilde{X} \times \tilde{Y} \mid f(x, y) \leq T\}$, for a given real-valued function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \mapsto \mathbb{R}$ and a cost target $T > 0$. Over the last years, optimization algorithms have received attention to deal with this class of problems [5, 7, 23, 31, 109], and some of the most famous are *branch-and-bound* [44] and *outer-approximation* [28, 37].

As presented in the next subsections, they are benchmarked against two derivative-based algorithms for solving 6 instances from a family of Power System Management problem. We just care to mention that the convergence analysis presented in Section 3.2 does not hold in this case because it depends on, essentially, the continuity of the stationarity measure π, defined in (3.7). The reason for that comes from the discontinuity of the orthogonal projection operator due to the presence of integer variables in the domain.

4.2.1 Solvers

In order to analyse the performance of the variants `TRCI` and `TRC` from Algorithm 1, we compare them with two other algorithms specialized to solve MINLP problems.

- `TRCI` and `TRC`: Variants of Algorithm 1 with the same input parameters, dictionary $\mathcal{D}_r$ and rules to update the interpolation set Y_{k+1}, as described in Section 4.1.1.

- `BONMIN`: Basic Open-source Nonlinear Mixed INteger programming algorithm[7] available in the `Matlab` OPTI Toolbox [21], with Outer Approximation set as the internal solver.

- `ELBM`: Extended Level-Bundle Method[8] from [24]. Similarly to `LB`, the objective function φ is replaced by the convex function $-\log(\varphi)$.

In both versions of Algorithm 1 we employed `BONMIN` to compute an approximate stationary point for the trust-region subproblem

[7] https://projects.coin-or.org/Bonmin

[8] We are grateful to Dr. Adriano R. Delfino, from UTFPR Brazil, for providing us the codes of the algorithm.

$$\max_{(x,y)\in X\times Y} \mathcal{M}^k(x,y) \quad \text{s.t.} \quad \|(x,y)-(\hat{x}^k,\hat{y}^k)\|_\diamond \leq \Delta_k. \tag{4.5}$$

where $(\hat{x}^k, \hat{y}^k)$ is the stability center at k-th iteration. Since in the test problems the discrete variables do not appear in the random inequality system $\xi \leq g(x)$, the random vector ξ follows a continuous multivariate Gaussian distribution. Consequently, the models $\mathcal{M}^k$, the probability function φ and its gradient $\nabla\varphi$ depend only on the continuous variable x, which means that φ and $\nabla\varphi$ can be evaluated by the same routine (`mvnqmc`) and number of simulations (10 000), as considered in the previous sections. In our tests, we fixed the maximum CPU time of 3600 seconds and increased the tolerance `tol` $= 10^{-4}$ for the stopping criteria of `TRCI` and `TRC`, while the other algorithms were set to default.

4.2.2 Test problem and numerical results

We consider 6 instances in a set of probability maximization problems, originally formulated as a maximization version of a MINLP CCP (see [24, Eq. 5.7]), where $f : \mathbb{R}^{n_x} \mapsto \mathbb{R}$ is a linear function and φ is defined as in (1.2), with $g : \mathbb{R}^{n_x} \mapsto \mathbb{R}^m$ a linear mapping. Following the ideas of the nonlinear continuous problems in Section 4.1, for a given initial point $(x^0, y^0) \in \tilde{X} \times \tilde{Y}$ and a parameter τ, we define the cost target by $T = \tau f(x^0)$ and reformulate the problems as probability maximization MINLP problems, as (4.4). Also, six uniformly spaced values of the parameter τ were set with the same condition as before and the initial point (x^0, y^0) is a solution of the simpler individual chance-constrained problem with mixed-integer variables

$$\begin{aligned} \underset{x,y}{\text{maximize}} \quad & f(x) \\ \text{subject to} \quad & \mathbb{P}\big[\xi_i \leq g_i(x)\big] \geq 0.95, \quad i = 1,\ldots,m \\ & (x,y) \in \tilde{X} \times \tilde{Y}. \end{aligned} \tag{4.6}$$

The information about the 6 instances of test problems are summarized in Table 4.6, similar to Table 4.2, but now the dimensions of the continuous and discrete parts (n_x and n_y) of the decision variable (x, y) are splitted and the last column shows all solvers

considered.

Type	Problem	n_x	n_y	m	Dist.	$\bar{t}$	τ	# inst.	Solvers
MINLP	Power Sys. Manag.	24	12	12	Gaussian	0.063	0.980 0.984 0.988 0.992 0.996 1.000	6	TRCI, TRC BONMIN ELBM

Table 4.6: Information about the test problems. Notation $\bar{t}$ stands for the estimated CPU time in seconds required to evaluate the objective function (oracle call).

Mixed-integer nonlinear programming problem

We consider one family of MINLP problems, where the random vector ξ follows a multivariate Gaussian distribution.

Power system management problem. This is an energy management problem from [24, Sec. 5.2][9], which consists on a short time planning period of two hydro power plants with reservoirs and a wind farm. The objective is to maximize the probability that the demands are satisfied while the profit by selling the leftover energy to the market reaches a minimal target, after attending the local community demand, the volumes of the reservoirs remain within bounds and at the end of the planning period the reservoirs levels must be greater or equal to a given level. In this example, we set a planning period of 12 hours. The decision variable $x \in \mathbb{R}^{n_x}$ ($n_x = 24$) corresponds to the energy produced by the hydro power plants, $y \in \{0,1\}^{n_y}$ ($n_y = 12$) models the turbines as "on/off" and the random vector $\xi \in \mathbb{R}^m$ ($m = 12$) corresponds to the energy generated by the wind farm.

Since we are benchmarking derivative-free and derivative-based solvers, we do not present data and performance profiles, only the results obtained by the algorithms by solving the MINLP instances, similar to the industrial problems. We keep using the criterion (4.2) to say when an algorithm solved a problem.

Table 4.7 reports the problem's data, the number of function and gradient evaluations and CPU time of the considered algorithms.

[9]We are grateful to Dr. Adriano R. Delfino, from UTFPR Brazil, for providing us the data of the problem.

Table 4.7: MINLP problems: computed function value, number of function and gradient evaluations, and CPU time spent by TRCI, TRC, BONMIN and ELBM.

Problem data	$\varphi(\hat{x}^k)$				#F				#G				CPU Time			
τ	TRCI	TRC	BONMIN	ELBM	TRCI	TRC	BONMIN	ELBM	TRCI	TRC	BONMIN	ELBM	TRCI	TRC	BONMIN	ELBM
0.980	0.993	0.993	0.993	0.993	163	16	1706	52	0	0	269	52	177.86	42.95	338.95	48.82
0.984	0.987	0.987	0.987	0.987	173	16	2760	49	0	0	385	49	168.15	28.26	510.16	45.33
0.988	0.977	0.977	0.976	0.976	161	15	1449	50	0	0	242	50	52.92	22.66	303.85	47.43
0.992	0.959	0.959	0.959	0.959	165	17	2228	46	0	0	363	46	63.37	31.65	439.05	40.94
0.996	0.933	0.933	0.933	0.933	156	16	2012	43	0	0	308	43	64.98	19.20	381.75	37.84
1.000	0.893	0.893	0.893	0.892	150	18	2311	43	0	0	337	43	58.04	21.38	433.87	37.47

From the numerical results we can observe that our approaches `TRCI` and `TRC` performed as good as the derivative-based algorithms specialized in solving MINLP problems: `BONMIN` and `ELBM`. There are some points that deserve attention: `TRC` used the least quantity of function evaluations, while `TRCI` used approximately 10 times, highlighting how expensive it is to update the interpolation set Y_{k+1}; even spending almost three times the number of function evaluations and computing the gradient of the objective function, `ELBM` did not spend twice the time of `TRC` to solve all instances, which means that solving the trust-region subproblem (4.5) was the most time consuming of our approach.

Summarizing the numerical results of this section, our approaches performed well, even though there is no guarantee of their convergence analysis for this class of problems. Also, we reinforce that `BONMIN` and `ELBM` are derivative-based algorithms, which can not be applied to probability distributions where the derivatives are not available.

Chapter 5

Conclusion

In this thesis, we proposed a derivative-free trust-region algorithm for probability maximization problems. The special structure of probability functions (whose derivatives are not available or are too expensive to be assessed) is exploited by easy-to-evaluate models that are linear combinations of copulæ with Lipschitz continuous gradients from a dictionary. Neither generalized concavity assumptions nor statistical work of copula estimation is necessary. Our algorithm updates the copula-based model at every iteration by solving a convex quadratic programming problem, which ensures that the model interpolates the probability function at least at the stability center. During the iterative process, the models capture the dependence structures between the marginal distributions of the probability function by assigning weights to the copulæ in the dictionary.

In each iteration of the algorithm, the subproblem consisting of minimizing the model in the trust region can be solved approximately: all is needed is a feasible point satisfying the efficiency condition (3.6). Under this assumption and mild hypotheses, the global convergence of the algorithm is presented ensuring that any accumulation point of the sequence generated by the algorithm is stationary.

Given the flexibility in constructing the models, two variants of the algorithm are proposed, namely `TRCI` and `TRC`. The first one is based on standard assumptions from the DFO literature and evaluates the probability function in at most $(n + 1)$ new points per

iteration, satisfying some geometric conditions. On the other hand, `TRC` requires only one function evaluation per iteration, ensures that the model is always a copula, but makes use of more stringent assumptions on the dictionary of copualæ (c.f. Section 3.2.3).

We assessed the numerical performance of these two variants on several instances of PMPs for solving four types of problems, being three with continuous and one with mixed-integer variables. For the continuous case, in which the global convergence is ensured, numerical comparisons with several state-of-art DFO solvers highlight the good performance of our approaches on the considered families of problems. The `TRC` was the only DFO method capable to deal with the large-scale industrial problems. Its economic rule to update the model allowed it to solve these problems in less than twenty four minutes, while well-known derivative-based methods either failed or took over one hour of processing. The good results motivated us to extend the numerical experiments to MINLP problems, where our approaches also performed well in comparison with two derivative-based algorithms. All the PMPs considered for benchmarking our proposal are log-concave, meaning that $\log(\varphi(\cdot))$ is a concave function. Although the log-transformation was not employed in our approach, but only in the derivative-based ones, our algorithm could compute (approximate) global solutions to all instances of the continuous and mixed-integer problems.

Concerning real-life applications of PMPs, our proposal enables practitioners to

- model uncertainties with more pertinent probability distributions, discarding the need for restricting their choices to a select group of distributions whose derivative formulæ are available and implementable;
- dismiss the non-trivial task of copula estimation (for the applications in which replacing the probability function with a copula is an option).

The advantages of the proposal, specified above, indicate that our DFO algorithm with copula-based models is a promising tool for dealing with probability maximization problems. This is evidenced in the numerical results when we compare our approach with other derivative-free trust-region algorithms with linear and quadratic models. Nonetheless, our proposal also has shortcomings that should be addressed in future research. For instance,

given a probability function φ, it is unclear to us how to certify that the dictionary of copulæ is diversified enough to ensure Assumption A3 for the `TRC` variant of Algorithm 1. In other words, we are not aware of how to ensure that the exact copula related to the distribution φ (according to Sklar's theorem) is included in the space spanned by the dictionary of copulæ. This is a theoretical subject of practical interest because it dismisses the need for having $(n + 1)$ interpolation points to build the model. A related question arises when the Algorithm 1 terminates for failure on line 6, which means that the dictionary is not rich enough to build a good model for φ and must be improved. A possible research direction consists of investigating how to learn from the model and function to select appropriate copulæ parameters to improve the dictionary. Another subject for future research is related to the global convergence analysis of the algorithm for solving MINLP problems, encouraged by the good numerical results. Since the stationarity measure is not continuous in this case, one possibility consists in fixing the discrete variables and then analyzing the behaviour of the continuous ones in an *outer-approximation* approach.

References

[1] C. Audet and W. Hare. *Derivative-Free and Blackbox Optimization*. Springer, 2017. DOI: 10.1007/978-3-319-68913-5.

[2] C. Audet and J. J. E. Dennis. "Mesh Adaptive Direct Search Algorithms for Constrained Optimization". In: *SIAM J. Optim.* 17.1 (2006), pp. 188–217. DOI: 10.1137/040603371.

[3] R. G. Bartle. *The Elements of Integration*. New York : Wiley, 1966.

[4] J. Y. Bello-Cruz, O. P. Ferreira and L. F. Prudente. "On the Global Convergence of the Inexact Semi-Smooth Newton Method for Absolute Value Equation". In: *Compu. Optim. Appl.* 65 (2016), pp. 93–108. DOI: 10.1007/s10589-016-9837-x.

[5] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke and A. Mahajan. "Mixed-Integer Nonlinear Optimization". In: *Acta Numer.* 22 (2013), pp. 1–131. DOI: 10.1017/S0962492913000032.

[6] A. S. Berahas, R. H. Byrd and J. Nocedal. "Derivative-Free Optimization of Noisy Functions via Quasi-Newton Methods". In: *SIAM J. Optim.* 29.2 (2019), pp. 965–993. DOI: 10.1137/18M1177718.

[7] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya and A. Wächter. "An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs". In: *Discrete Optim.* 5.2 (2008), pp. 186–204. DOI: 10.1016/j.disopt.2006.10.011.

[8] Z. I. Botev. "The Normal Law Under Linear Restrictions: Simulation and Estimation via Minimax Tilting". In: *J. Roy. Stat. Soc. B.* 79.1 (2017), pp. 125 –148. DOI: 10.1111/rssb.12162.

[9] L. F. Bueno, A. Friedlander, J. M. Martínez and F. N. C. Sobral. "Inexact Restoration Method for Derivative-Free Optimization with Smooth Constraints". In: *SIAM J. Optim.* 23.2 (Jan. 2013), pp. 1189–1213. DOI: 10.1137/110856253.

[10] E. Butyn, E. W. Karas and W. de Oliveira. "A Derivative-Free Trust-Region Algorithm with Copula-Based Models for Probability Maximization Problems". In: *Eur. J. Oper. Res.* (2021). DOI: 10.1016/j.ejor.2021.09.040.

[11] S. Cambanis, S. Huang and G. Simons. "On the Theory of Elliptically Contoured Distributions". In: *J. Multivariate Anal.* 11.3 (1981), pp. 368–385. DOI: 10.1016/0047-259X(81)90082-8.

[12] B. Colson and P. L. Toint. "Exploiting Band Structure in Unconstrained Optimization Without Derivatives". In: *Optim. Eng.* 2 (2001), pp. 399–412. DOI: 10.1023/A:1016090421852.

[13] B. Colson and P. L. Toint. "Optimizing Partially Separable Functions without Derivatives". In: *Optim. Method. Softw.* 20.4-5 (2005), pp. 493–508. DOI: 10.1080/10556780500140227.

[14] P. D. Conejo, E. W. Karas and L. G. Pedroso. "A Trust-Region Derivative-Free Algorithm for Constrained Optimization". In: *Optim. Method Softw.* 30.6 (2015), pp. 1126–1145. DOI: 10.1080/10556788.2015.1026968.

[15] P. D. Conejo, E. W. Karas, L. G. Pedroso, A. A. Ribeiro and M. Sachine. "Global Convergence of Trust-Region Algorithms for Convex Constrained Minimization without Derivatives". In: *Appl. Math. Comput.* 220.1 (2013), pp. 324–330. DOI: 10.1016/j.amc.2013.06.041.

[16] A. R. Conn, N. I. M. Gould and P. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. Philadelphia: SIAM, 2000. DOI: 10.1137/1.9780898719857.

[17] A. R. Conn, K. Scheinberg and P. L. Toint. "On the Convergence of Derivative-Free Methods for Unconstrained Optimization". In: *Approximation Theory and Optimization: Tributes to M. J. D. Powell*. Ed. by A. Iserles and M. Buhmann. Vol. 7. Cambridge University Press, 1997, pp. 83–108.

[18] A. R. Conn, K. Scheinberg and L. N. Vicente. "Global Convergence of General Derivative-Free Trust-Region Algorithms to First and Second Order Critical Points". In: *SIAM J. Optim.* 20.1 (2009), pp. 387–415. DOI: 10.1137/060673424.

[19] A. R. Conn, K. Scheinberg and L. N. Vicente. *Introduction to Derivative-Free Optimization*. Series on Optimization. MOS - SIAM, 2009. DOI: 10.1137/1.9780898718768.

[20] A. R. Conn and P. L. Toint. "An Algorithm Using Quadratic Interpolation for Unconstrained Derivative Free Optimization". In: *Di Pillo G., Giannessi F. (eds) Nonlinear Optimization and Applications* (1996), pp. 27–47. DOI: 10.1007/978-1-4899-0289-4_3.

[21] J. Currie and D. I. Wilson. "OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User". In: *Foundations of Computer-Aided Process Operations*. Ed. by N. Sahinidis and J. Pinto. Savannah, Georgia, USA, Jan. 2012.

[22] W. de Oliveira. "Proximal Bundle Methods for Nonsmooth DC Programming". In: *J. Global. Optim.* 75 (2019), pp. 523–563. DOI: 10.1007/s10898-019-00755-4.

[23] W. de Oliveira. "Regularized Optimization Methods for Convex MINLP Problems". In: *TOP* 24 (2016), pp. 665–692. DOI: 10.1007/s11750-016-0413-4.

[24] A. R. Delfino. "Outer-approximation algorithms for nonsmooth convex MINLP problems with chance constrains". PhD thesis. Federal University of Paraná, 2018.

[25] A. R. Delfino. "Solving an MINLP with Chance Constraint Using a Zhang's Copula Family". In: *Optimization of Complex Systems: Theory, Models, Algorithms and Applications. WCGO 2019*. Ed. by H. Le Thi, H. M. Le and T. Pham Dinh. Vol. 991.

Advances in Intelligent Systems and Computing. Springer, 2020, pp. 477–487. DOI: 10.1007/978-3-030-21803-4_48.

[26] S. L. Digabel. "NOMAD: Nonlinear Optimization with the MADS Algorithm". In: *ACM T. Math. Software* 37.4 (2011), p. 44. DOI: 10.1145/1916461.1916468.

[27] E. D. Dolan and J. J. Moré. "Benchmarking Optimization Software with Performance Profiles". In: *Math. Program* 91 (2009), pp. 201–213. DOI: 10.1007/s101070100263.

[28] M. A. Duran and I. E. Grossmann. "An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs". In: *Math. Program.* 36 (1986), pp. 307–339. DOI: 10.1007/BF02592064.

[29] F. Durante and C. Sempi. *Principles of Copula Theory*. New York: Chapman and Hall/CRC, 2015. DOI: 10.1201/b18674.

[30] R. Durrett. *Probability: Theory and Examples*. Cambridge University Press, 2010.

[31] V.-P. Eronen, M. M. Mäkelä and T. Westerlund. "On the Generalization of ECP and OA Methods to Nonsmooth Convex MINLP Problems". In: *Optimization* 63.7 (2014), pp. 1057–1073. DOI: 10.1080/02331934.2012.712118.

[32] C. I. Fabian, E. Csizmás, R. Drenyovszki, W. van Ackooij, T. Vajnai, L. Kovács and T. Szántai. "Probability Maximization by Inner Approximation". In: *Acta Polytech. Hung.* 15.1 (2018), pp. 105–125. DOI: 10.12700/APH.15.1.2018.1.7.

[33] K. Fang, S. Kotz and K. W. Ng. *Symmetric Multivariate and Related Distributions*. 1st edn. Monographs on statistics and applied probability. New York: Chapman and Hall, 1990.

[34] G. Fasano, J. L. Morales and J. Nocedal. "On the Geometry Phase in Model-Based Algorithms for Derivative-Free Optimization". In: *Optim. Method. Softw.* 24.1 (2009), pp. 145–154. DOI: 10.1080/10556780802409296.

[35] P. S. Ferreira, E. W. Karas, M. Sachine and F. N. C. Sobral. "Global Convergence of a Derivative-Free Inexact Restoration Filter Algorithm for Nonlinear Programming". In: *Optimization* 66.2 (2017), pp. 271–292. DOI: 10.1080/02331934.2016.1263629.

[36] P. S. Ferreira, E. W. Karas and M. Sachine. "A Globally Convergent Trust-Region Algorithm for Unconstrained Derivative-Free Optimization". In: *Comp. Appl. Math.* 34 (2015), pp. 1075–1103. DOI: 10.1007/s40314-014-0167-2.

[37] R. Fletcher and S. Leyffer. "Solving Mixed Integer Nonlinear Programs by Outer Approximation". In: *Math. Program.* 66 (1994), pp. 327–349. DOI: 10.1007/BF01581153.

[38] M. Frank and P. Wolfe. "An Algorithm for Quadratic Programming". In: *Naval Res. Logis. Quart.* 3 (1956), pp. 95–110. DOI: 10.1002/nav.3800030109.

[39] P. Gilmore and C. T. Kelley. "An Implicit Filtering Algorithm for Optimization of Functions with Many Local Minima". In: *SIAM J. Optim.* 5.2 (1995), pp. 269–285. DOI: 10.1137/0805015.

[40] C. C. Gonzaga, E. W. Karas and M. Vanti. "A Globally Convergent Filter Method for Nonlinear Programming". In: *SIAM J. Optimization* 14.3 (2003), pp. 646–669. DOI: 10.1137/S1052623401399320.

[41] A. A. Gouda and T. Szántai. "On Numerical Calculation of Probabilities According to Dirichlet Distribution". In: *Ann. Oper. Res.* 177 (2010), pp. 185–200. DOI: 10.1007/s10479-009-0601-9.

[42] G. N. Grapiglia, J. Yuan and Y.-X. Yuan. "A Derivative-Free Trust-Region Algorithm for Composite Nonsmooth Optimization". In: *Comp. Appl. Math.* 35 (2016), pp. 475–499. DOI: 10.1007/s40314-014-0201-4.

[43] S. Gratton, P. L. Toint and A. Tröltzsch. "An Active-Set Trust-Region Method for Derivative-Free Nonlinear Bound-Constrained Optimization". In: *Optim. Method. Softw.* 26.4-5 (2011), pp. 873–894. DOI: 10.1080/10556788.2010.549231.

[44] O. K. Gupta and A. Ravindran. "Branch and Bound Experiments in Convex Nonlinear Integer Programming". In: *Manage. Sci.* 31.12 (1985), pp. 1533–1546. DOI: 10.1287/mnsc.31.12.1533.

[45] W. Hare. "A Discussion on Variational Analysis in Derivative-Free Optimization". In: *Set-Valued Var. Anal.* 28 (2020), pp. 643–659. DOI: 10.1007/s11228-020-00556-y.

[46] W. Hare, C. Planiden and C. Sagastizábal. "A Derivative-Free VU-Algorithm for Convex Finite-Max Problems". In: *Optim. Method. Softw.* 35.3 (Sept. 2019), pp. 521–559. DOI: 10.1080/10556788.2019.1668944.

[47] R. Henrion and A. Möller. "Optimization of a Continuous Distillation Process under Random Inflow Rate". In: *Comput. Math. Appl.* 45 (2003), pp. 247–262. DOI: 10.1016/S0898-1221(03)80017-2.

[48] R. Henrion. "Introduction to Chance Constraint Programming". In: *Tutorial paper for the Stochastic Programming Community Home Page* (2004). URL: http://www.wias-berlin.de/people/henrion/ccp.ps.

[49] R. Henrion and A. Möller. "A Gradient Formula for Linear Chance Constraints under Gaussian Distribution". In: *Math. Oper. Res.* 37 (2012), pp. 475–488. DOI: 10.1287/moor.1120.0544.

[50] R. Hooke and T. A. Jeeves. "Direct Search Solution of Numerical and Statistical Problems". In: *J. Assoc. Comput. Mach.* 8.2 (1961), pp. 212–229. DOI: 10.1145/321062.321069.

[51] T. G. Kolda, R. M. Lewis, and V. Torczon. *A Generating Set Direct Search Augmented Lagrangian Algorithm for Optimization with a Combination of General and Linear Constraints*. Tech. rep. United States: Office of Scientific and Technical Information, 2006. DOI: 10.2172/893121.

[52] T. G. Kolda, R. M. Lewis, and V. Torczon. "Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods". In: *SIAM Rev.* 45.3 (2003), pp. 385–482. DOI: 10.1137/S003614450242889.

[53] Z. M. Landsman and E. A. Valdez. "Tail Conditional Expectations for Elliptical Distributions". In: *North American Actuarial Journal* 7.4 (2013), pp. 55–71. DOI: 10.1080/10920277.2003.10596118.

[54] J. Larson, M. Menickelly and S. M. Wild. "Derivative-Free Optimization Methods". In: *Acta Numer.* 28 (2019), pp. 287–404. DOI: 10.1017/S0962492919000060.

[55] C. Lemaréchal, A. Nemirovskii and Y. Nesterov. "New Variants of Bundle Methods". In: *Math. Program.* 69 (1995), pp. 111–147. DOI: 10.1007/BF01585555.

[56] R. M. Lewis and V. Torczon. "Pattern Search Algorithms for Bound Constrained Minimization". In: *SIAM J. Optim.* 9.4 (1999), pp. 1082–1099. DOI: 10.1137/S1052623496300507.

[57] R. M. Lewis and V. Torczon. "Pattern Search Methods for Linearly Constrained Minimization". In: *SIAM J. Optim.* 10.3 (2000), pp. 917–941. DOI: 10.1137/S1052623497331373.

[58] J. Luedtke, S. Ahmed and G. L. Nemhauser. "An Integer Programming Approach for Linear Programs with Probabilistic Constraints". In: *Math. Program.* 122 (2010), pp. 247–272. DOI: 10.1007/s10107-008-0247-4.

[59] J. Luedtke and S. Ahmed. "A Sample Approximation Approach for Optimization with Probabilistic Constraints". In: *SIAM J. Optim.* 19.2 (2008), pp. 674–699. DOI: 10.1137/070702928.

[60] J. M. Martínez and A. C. Moretti. "A Trust-Region Method for Minimization of Nonsmooth Functions with Linear Constraints". In: *Math. Program.* 76 (1997), pp. 431–449. DOI: 10.1007/BF02614392.

[61] J. J. Moré and S. Wild. "Benchmarking Derivative-Free Optimization Algorithms". In: *SIAM J. Optim.* 20.1 (2009), pp. 172–191. DOI: 10.1137/080724083.

[62] J. A. Nelder and R. Mead. "A Simplex Method for Function Minimization". In: *Comput. J.* 7.4 (1965), pp. 308–313. DOI: 10.1093/comjnl/7.4.308.

[63] R. B. Nelsen. *An Introduction to Copulas*. 2nd. New York: Springer, 2006. DOI: 10.1007/0-387-28678-0.

[64] Y. Nesterov. *Introductory Lectures on Convex Optimization: a Basic Course*. Vol. 87. Boston, Dordrecht, London: Kluwer Academic, 2004. DOI: 10.1007/978-1-4419-8853-9.

[65] J. Nocedal and S. J. Wright. *Numerical Optimization*. 2nd. Springer Series in Oper. Res. Springer-Verlag, 2006. DOI: 10.1007/978-0-387-40065-5.

[66] D. Paindaveine. *Elliptical symmetry*. Encyclopedia of Environmetrics, 2012.

[67] G. A. Periçaro, A. A. Ribeiro and E. W. Karas. "Global Convergence of a General Filter Algorithm Based on an Efficiency Condition of the Step." In: *Applied Mathematics and Computation* 219(17) (2013), pp. 9581–9597. DOI: 10.1016/j.amc.2013.03.012.

[68] M. J. D. Powell. "A Direct Search Optimization Method that Models the Objective and Constraint Functions by Linear Interpolation". In: *Advances in Optimization and Numerical Analysis*. Ed. by S. Gomez and J. P. Hennart. Vol. 275. Mathematics and Its Applications. Springer, 1994, pp. 51–67. DOI: 10.1007/978-94-015-8330-5_4.

[69] M. J. D. Powell. "On fast trust region methods for quadratic models with linear constraints". In: 7 (2015), pp. 237–267. DOI: 10.1007/s12532-015-0084-4.

[70] M. J. D. Powell. "On the Convergence of Trust Region Algorithms for Unconstrained Minimization without Derivatives". In: *Comput. Optim. Appl.* 53 (2012), pp. 527–555. DOI: 10.1007/s10589-012-9483-x.

[71] M. J. D. Powell. *The BOBYQA Algorithm for Bound Constrained Optimization without Derivatives*. Tech. rep. Department of Applied Mathematics and Theoretical Physics, University of Cambridge, 2009.

[72] M. J. D. Powell. "The NEWUOA Software for Unconstrained Optimization without Derivatives". In: *Large-Scale Nonlinear Optimization*. Ed. by G. Di Pillo and M. Roma. Vol. 83. Nonconvex Optimization and its Applications. Springer, 2006, pp. 255–297. DOI: 10.1007/0-387-30065-1_16. URL: https://scholar.google.com/scholar?cluster=12733354653570047722.

[73] M. Powell. "On Derivative-Free Optimization with Linear Constraints". In: *21st ISMP* (2012).

[74] A. Prékopa. "Programming under Probabilistic Constraint and Maximizing Probabilities under Constraints". In: *Stochastic Programming. Mathematics and Its Applications*. Vol. 324. Dordrecht: Springer, 1995, pp. 319 –371. DOI: 10.1007/978-94-017-3087-7_11.

[75] A. Prékopa and T. Szántai. "A New Multivariate Gamma Distribution and its Fitting to Empirical Streamflow Data". In: *Water Resour. Res.* 14.1 (1978), pp. 19–24. DOI: 10.1029/WR014i001p00019.

[76] A. Prékopa, S. Ganczer, I. Deák and K. Patyi. "The STABIL Stochastic Programming Model and its Experimental Application to the Electrical Energy Sector of the Hungarian Economy". In: *Stochastic Programming*. Ed. by M. Dempster. Academic Press, 1980, pp. 369–385.

[77] A. Prékopa. *Stochastic Programming*. Mathematics and Its Applications. Netherlands: Springer, 1995. DOI: 10.1007/978-94-017-3087-7.

[78] A. Prékopa. "On Probabilistic Constrained Programming". In: *Kuhn, H. (ed.) Proceedings of the Princeton Symposium on Math. Prog.* vol. 28 (1970), pp. 113–138. DOI: 10.1515/9781400869930-009.

[79] T. M. Ragonneau and Z. Zhang. "PDFO: Cross-Platform Interfaces for Powell's Derivative-Free Optimization Solvers (Version 1.1)". In: (2021). DOI: 10.5281/zenodo.3887569. URL: https://www.pdfo.net.

[80] L. M. Rios and N. V. Sahinidis. "Derivative-Free Optimization: a Review of Algorithms and Comparison of Software Implementations". In: *J. Glob. Optim.* 56 (2013), pp. 1247–1293. DOI: 10.1007/s10898-012-9951-y.

[81] R. T. Rockafellar and R. J. B. Wets. *Variational Analysis*. 3rd. Vol. 317. Grundlehren der mathematischen Wissenschaften. Berlin, Heidelberg: Springer, 2009. DOI: https://doi.org/10.1007/978-3-642-02431-3.

[82] K. Scheinberg and P. L. Toint. "Self-Correcting Geometry in Model-Based Algorithms for Derivative-Free Unconstrained Optimization". In: *SIAM J. Optim.* 20.6 (2010), pp. 3512–3532. DOI: 10.1137/090748536.

[83] A. Shapiro, D. Dentcheva and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM Series on Optimization. Philadelphia: SIAM, 2009. DOI: 10.1137/1.9780898718751.

[84] N. P. Stamatatou. "Bivariate frequency analysis of extreme rainfall and floods using copulas". MA thesis. University of Thessaly, 2017.

[85] P. L. Toint. "Global Convergence of a Class of Trust-Region Methods for Nonconvex Minimization in Hilbert Space". In: *IMA J. Numer. Anal.* 8.2 (1988), pp. 231–252. DOI: 10.1093/imanum/8.2.231.

[86] V. Torczon. "On the Convergence of Pattern Search Algorithms". In: *SIAM J. Optim.* 7.1 (1997), pp. 1–25. DOI: 10.1137/S1052623493250780.

[87] A. Tröltzsch. "An Active-set Trust-Region Method for Bound-Constrained Nonlinear Optimization without Derivatives Applied to Noisy Aerodynamic Design Problems". PhD thesis. Université de Toulouse, 2011.

[88] S. Uryasev. "Derivatives of Probability and Integral Functions: General Theory and Examples". In: *Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization. 2nd edn.* (2009). Ed. by C. A. Floudas, pp. 658–663. DOI: 10.1007/978-0-387-74759-0_119.

[89] S. Uryasev. "Derivatives of Probability Functions and Integrals Over Sets Given by Inequalities". In: *J. Comput. Appl. Math.* 56.1-2 (1994), pp. 197–223. DOI: 10.1016/0377-0427(94)90388-3.

[90] S. Uryasev. "Derivatives of Probability Functions and Some Applications". In: *Ann. Oper. Res.* 56 (1995), pp. 287–311. DOI: 10.1007/BF02031712.

[91] S. Uryasev. "Introduction to the Theory of Probabilistic Functions and Percentiles (Value-At-Risk)". In: *Uryasev S.P. (eds) Probabilistic Constrained Optimization. Nonconvex Optimization and Its Applications* 49 (2000), pp. 1–25. DOI: 10.1007/978-1-4757-3150-7_1.

[92] W. van Ackooij. "A Discussion of Probability Functions and Constraints from a Variational Perspective". In: *Set-Valued Var. Anal.* 28 (2020), pp. 585–609. DOI: 10.1007/s11228-020-00552-2.

[93] W. van Ackooij. "Chance constrained programming with applications in energy management". PhD thesis. École Centrale des Arts et Manufactures, 2013.

[94] W. van Ackooij, I. Aleksovska and M. Munoz-Zuniga. "(Sub-)Differentiability of Probability Functions with Elliptical Distributions". In: *Set-Valued Var. Anal.* 26 (2018), pp. 887–910. DOI: 10.1007/s11228-017-0454-3.

[95] W. van Ackooij and W. de Oliveira. "Convexity and Optimization with Copulae Structured Probabilistic Constraints". In: *Optim. J. Math. Program. Oper. Res.* 65.7 (2016), pp. 1349–1376. DOI: 10.1080/02331934.2016.1179302.

[96] W. van Ackooij and W. de Oliveira. "Level Bundle Methods for Constrained Convex Optimization with Various Oracles". In: *Computational Optimization and Applications* 57 (2014), pp. 555–597. DOI: 10.1007/s10589-013-9610-3.

[97] W. van Ackooij and W. de Oliveira. "Nonsmooth and Nonconvex Optimization Via Approximate Difference-of-Convex Decompositions". In: *J. Optim. Theory App.* 182 (2019), pp. 49–80. DOI: 10.1007/s10957-019-01500-3.

[98] W. van Ackooij, E. C. Finardi and G. M. Ramalho. "An Exact Solution Method for the Hydrothermal Unit Commitment Under Wind Power Uncertainty With Joint Probability Constraints". In: *IEEE T. Power. Syst.* 33.6 (2018), pp. 6487–6500. DOI: 10.1109/TPWRS.2018.2848594.

[99] W. van Ackooij and R. Henrion. "Gradient Formulae for Nonlinear Probabilistic Constraints with Gaussian and Gaussian-Like Distributions". In: *SIAM J. Optim.* 24.4 (2014), pp. 1864–1889. DOI: 10.1137/130922689.

[100] W. van Ackooij and R. Henrion. "(Sub-)Gradient Formulae for Probability Functions of Random Inequality Systems Under Gaussian Distribution". In: *SIAM/ASA J. Uncertainty Quantification* 5.1 (Jan. 2017), pp. 63–87. DOI: 10.1137/16M1061308.

[101] W. van Ackooij and J. Malick. "Eventual Convexity of Probability Constraints with Elliptical Distributions". In: *Math. Program.* 175 (2019), pp. 1–27. DOI: 10.1007/s10107-018-1230-3.

[102] W. van Ackooij and J. Malick. "Second-Order Differentiability of Probability Functions". In: *Optim. Lett.* 11 (2017), pp. 179–194. DOI: 10.1007/s11590-016-1015-7.

[103] W. van Ackooij and C. Sagastizábal. "Constrained Bundle Methods for Upper Inexact Oracles with Application to Joint Chance Constrained Energy Problems". In: *SIAM J. Optim.* 24.2 (2014), pp. 733–765. DOI: 10.1137/120903099.

[104] W. van Ackooij, S. Demassey, P. Javal, H. Morais, W. de Oliveira and B. Swaminathan. "A Bundle Method for Nonsmooth DC Programming with Application to Chance-Constrained Problems". In: *Comput. Optim. Appl.* 78 (Nov. 2021), pp. 451–490. DOI: 10.1007/s10589-020-00241-8.

[105] W. van Ackooij, R. Henrion, A. Möller and R. Zorgati. "Joint Chance Constrained Programming for Hydro Reservoir Management". In: *Optim. Eng.* 15 (2014), pp. 509–531. DOI: 10.1007/s11081-013-9236-4.

[106] W. van Ackooij, R. Henrion, A. Möller and R. Zorgati. "On Probabilistic Constraints Induced by Rectangular Sets and Multivariate Normal Distributions". In: *Math. Meth. Oper. Res.* 71.3 (2010), pp. 535–549. DOI: 10.1007/s00186-010-0316-3.

[107] A. I. F. Vaz and L. N. Vicente. "A Particle Swarm Pattern Search Method for Bound Constrained Global Optimization". In: *J. Global Optim.* 39 (2007), pp. 197–219. DOI: 10.1007/s10898-007-9133-5.

[108] A. Verdério, E. W. Karas, L. G. Pedroso and K. Scheinberg. "On the Construction of Quadratic Models for Derivative-Free Trust-Region Algorithms". In: *EURO J. Comput. Optim.* 5.4 (2017), pp. 501–527. DOI: 10.1007/s13675-017-0081-7.

[109] Z. Wei and M. M. Ali. "Convex Mixed Integer Nonlinear Programming Problems and an Outer Approximation Algorithm". In: *J. Glob. Optim.* 63 (2015), pp. 213–227. DOI: 10.1007/s10898-015-0284-5.

[110] S. M. Wild, R. G. Regis and C. A. Shoemaker. "ORBIT: Optimization by Radial Basis Function Interpolation in Trust-Regions". In: *SIAM J. Sci. Comput.* 30.6 (2008), pp. 3197–3219. DOI: 10.1137/070691814.

[111] S. M. Wild and C. A. Shoemaker. "Global Convergence of Radial Basis Function Trust-Region Algorithms for Derivative-Free Optimization". In: *SIAM Rev.* 55.2 (2013), pp. 349–371. DOI: 10.1137/120902434.

[112] S. M. Wild. "Derivative-free optimization algorithms for computationally expensive functions". PhD thesis. Cornell University, Ithaca, NY, 2008.

[113] S. S. Wilks. *Mathematical Statistics*. John Wiley & Sons, 1962.

[114] M. Xi, W. Sun and J. Chen. "Survey of Derivative-Free Optimization". In: *SIAM J. Optim.* 10.4 (2020), pp. 537–555. DOI: 10.3934/naco.2020050.

[115] L. Zhang, W. Zhou and D. Li. "Global Convergence of a Modified Fletcher-Reeves Conjugate Gradient Method with Armijo-type Line Search". In: *Numer. Math.* 104 (2006), pp. 561–572. DOI: 10.1007/s00211-006-0028-z.