UNIVERSIDADE FEDERAL DO PARANÁ

ADRIANO RODRIGO DELFINO

OUTER-APPROXIMATION ALGORITHMS FOR NONSMOOTH CONVEX MINLP PROBLEMS WITH CHANCE CONSTRAINTS

CURITIBA
2018

ADRIANO RODRIGO DELFINO

OUTER-APPROXIMATION ALGORITHMS FOR NONSMOOTH CONVEX MINLP PROBLEMS WITH CHANCE CONSTRAINTS

Tese apresentada ao curso de Pós-Graduação em Matemática da Universidade Federal do Paraná, como requisito parcial à obtenção do título de Doutor em Matemática.
Orientador: Prof. Dr. Yuan Jin Yun
Coorientador: Prof. Dr. Welington Luis de Oliveira

CURITIBA
2018

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

D349o Delfino, Adriano Rodrigo
Outer-approximation algorithms for nonsmooth convex MINLP problems with chance constraints / Adriano Rodrigo Delfino. – Curitiba, 2018.

Tese - Universidade Federal do Paraná, Setor de Matemática , Programa de Pós-Graduação em Ciências Exatas, 2018.

Orientador: Yuan Jin Yun – Coorientador: Welington Luis de Oliveira.

1. Otimização matemática. 2. Equações diferenciais estocásticas. 3. Probabilidades. I. Universidade Federal do Paraná. II. Yun, Yuan Jin. III. Oliveira, Welington Luis de. IV. Título.

CDD: 519.3

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

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

ATA Nº022

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

No dia treze de abril de dois mil e dezoito às 09:30 horas, na sala PA 300, Rua Cel. Francisco H. dos Santos, 100 - Jardim das Américas, foram instalados os trabalhos de arguição do doutorando **ADRIANO RODRIGO DELFINO** para a Defesa Pública de sua tese intitulada **Outer-approximation algorithms for nonsmooth convex MINLP problems with chance constraints**. A Banca Examinadora, designada pelo Colegiado do Programa de Pós-Graduação em MATEMÁTICA da Universidade Federal do Paraná, foi constituída pelos seguintes Membros: ELIZABETH WEGNER KARAS (UFPR), ADEMIR ALVES RIBEIRO (UFPR), ROGER BEHLING (UFSC), JOSÉ ALBERTO RAMOS FLOR (UFPR), PAULO J. SILVA E SILVA (UNICAMP), . Dando início à sessão, a presidência passou a palavra ao discente, para que o mesmo expusesse seu trabalho aos presentes. Em seguida, a presidência passou a palavra a cada um dos Examinadores, para suas respectivas arguições. O aluno respondeu a cada um dos arguidores. A presidência retomou a palavra para suas considerações finais. A Banca Examinadora, então, reuniu-se e, após a discussão de suas avaliações, decidiu-se pela APROVAÇÃO do aluno. O doutorando foi convidado a ingressar novamente na sala, bem como os demais assistentes, após o que a presidência fez a leitura do Parecer da Banca Examinadora. A aprovação no rito de defesa deverá ser homologada pelo Colegiado do programa, mediante o atendimento de todas as indicações e correções solicitadas pela banca dentro dos prazos regimentais do programa. A outorga do 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, ELIZABETH WEGNER KARAS, lavrei a presente ata, que vai assinada por mim e pelos membros da Comissão Examinadora.

Curitiba, 13 de Abril de 2018.

ELIZABETH WEGNER KARAS
Presidente da Banca Examinadora

ADEMIR ALVES RIBEIRO
Avaliador Interno

ROGER BEHLING
Avaliador Externo

JOSÉ ALBERTO RAMOS FLOR
Avaliador Externo

PAULO J. SILVA E SILVA
Avaliador Externo

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

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em MATEMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **ADRIANO RODRIGO DELFINO** intitulada: **Outer-approximation algorithms for nonsmooth convex MINLP problems with chance constraints**, após terem inquirido o aluno e realizado 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, 13 de Abril de 2018.

ELIZABETH WEGNER KARAS
Presidente da Banca Examinadora

ADEMIR ALVES RIBEIRO
Avaliador Interno

ROGER BEHLING
Avaliador Externo

JOSÉ ALBERTO RAMOS FLOR
Avaliador Externo

PAULO J. SILVA E SILVA
Avaliador Externo

To my family.

ACKNOWLEDGMENTS

Firstly, I would like to express my gratefulness to God. Without His support, I would not achieve anything so far.

A very special gratitude goes out to my advisors Yuan Jin Yun and Welington de Oliveira. They provided me great support in these past years and passed me a lot of knowledge which I will carry on for my whole life.

I am eternally grateful to my family. My parents, Sonia and Jose, have always believed in me and have provided me through moral and emotional support in my life. A special gratitude to my brothers, Alessandro and Alisson, who have been present in my life and helped me on tough times. They always were there to encourage me to not give up.

I would like to thank my fellow doctoral students, especially Teles, Claudio Avila and Joseane, for their feedback, cooperation and of course friendship. In addition I would like to express my gratitude to great friends Alberto, Jaqueline, Ronaldo and Gilberto whom I have special consideration.

My special token of appreciation goes to the Federal University of Paraná, which was a place where I spent ten worthy years learning every day something new. I am also grateful to all the teachers who somehow participated in this journey. I also place on record, my sense of gratitude to the Federal University of Technology of Paraná that has been my second home since 2010.

Last but not the least, I would like to thank my thesis committee for their encouragement, insightful comments, and hard questions which was a valuable contribution to this work.

"We do not know what we want
and yet we are responsible for
what we are - that is the fact."

Jean-Paul Sartre

RESUMO

As restrições de probabilidade desempenham um papel fundamental nos problemas de otimização envolvendo incertezas. Essas restrições exigem que um sistema de desigualdade dependendo de um vetor aleatório tenha que ser satisfeito com uma probabilidade suficientemente alta. Neste trabalho, lidamos com problemas de otimização com restrições de probabilidades envolvendo variáveis inteiras. Assumimos que as funções envolvidas são convexas e a restrição de probabilidade tenha propriedade generalizada de convexidade. Para lidar com problemas de otimização desse tipo, combinamos o algoritmo de aproximação externa (OA) e o algoritmo de feixes. Os algoritmos OA tem sido aplicado para problemas suáveis e para uma pequena classe limitada de problemas não-suáveis. Neste trabalho, estendemos o algoritmo OA para lidar com problemas mais gerais não-suáveis. Além disso, mostramos que quando os subproblemas não-lineares resultantes do algoritmo OA são resolvidos por um método de feixes, então os subgradientes que satisfazem as condições de Karush Kuhn Tucker (KKT) estão prontamente disponíveis independentemente da estrutura das funções convexas não-suáveis. Esta propriedade é crucial para provar a convergência (finita) do algoritmo OA. Problemas com restrições probabilísticas aparecem, por exemplo, em modelos de energia (estocásticos). No contexto de interesse, pelo menos uma das restrições não lineares envolve uma função de probabilidade $P[h(x,y) \geq \xi]$, onde h é uma função côncava e $\xi \in \mathbb{R}^m$ é um vetor aleatório. Em geral, uma integração numérica multidimensional é empregada para avaliar essa função de probabilidade. Como uma alternativa para lidar com restrições de probabilidades (que é muito cara computacionalmente), propomos a aproximação da medida de probabilidade P por uma cópula apropriada. Nós investigamos uma família de cópulas não-suáveis e fornecemos algumas propriedades generalizadas de convexidade novas e úteis. Em particular, provamos que a família de cópulas de Zhang é $\alpha-$côncava para todo $\alpha \leq 0$. Esse resultado nos permite aproximar as restrições probabilísticas por restrições muito mais simples envolvendo cópulas. Avaliamos numericamente as abordagens dadas em duas classe de problemas provenientes do gerenciamento do sistema de energia elétrica.

Palavras-chave: *Otimização não-linear inteira, Otimização Estocástica, Restrições Probabilísticas.*

ABSTRACT

Probability constraints play a key role in optimization problems involving uncertainties. These constraints (also known as chance constraints) require that an inequality system depending on a random vector has to be satisfied with high enough probability. In this work we deal with chance-constrained optimization problems having mixed-integer variables. We assume that the involved functions are convex and the probability constraint has generalized convexity properties. In order to deal with optimization problems of this type, we combine outer-approximation (OA) and bundle method algorithms. OA algorithms have been applied to smooth problems and to a small class of nonsmooth problems. In this work we extend the OA to handle more general nonsmooth problems. Moreover, we show that when the resulting OA's nonlinear subproblems are solved by a bundle method, then subgradients satisfying the Karush-Kuhn-Tucker (KKT) conditions are readily available regardless the structure of the nonsmooth convex functions. This property is crucial for proving (finite) convergence of the OA algorithm. Chance-constrained problems appear, for instance, in (stochastic) energy models. In the context of interest, at least one nonlinear constraint models the probability function $P[h(x, y) \geq \xi]$, where h is a concave map and $\xi \in \mathbb{R}^m$ is a random vector. In general, multidimensional numerical integration is employed to evaluate this probability function. As an alternative to deal with probability constraints (which is very expensive computationally), we propose approximating the probability measure P with a suitable copula. We investigate a family of nonsmooth copulae and provide some new and useful generalized convexity properties. In particular, we prove that Zhang's copulae are α-concave for all $\alpha \leq 0$. This result allows us to approximate chance-constrained programs by much simpler copula-constrained ones. We assess numerically the given approaches on two classes of problems coming from power system management.

Keywords: *Mixed-Integer Nonlinear Optimization, Stochastic Optimization, Chance constraints.*

Contents

Introduction

Many real-life optimization problems are modeled in a mixed-integer setting, involving discrete and continuous decision variables. Optimization algorithms for solving *mixed-integer nonlinear programming* (MINLP) problems have become an important focus of research over the last years [4, 9, 16, 26, 32, 35, 75, 77]. Most of these algorithms require the involved functions to be convex and differentiable. The latter hypothesis is not assumed in this work. Moreover, we are particularly interested in nonsmooth convex MINLP problems that have at least one probability constraint. Problems of this type fall into a very challenging class of optimization problems, because not only smoothness and convexity (of the feasible set) are absent, but also the probability constraint is in general difficult to be evaluated, because it requires computing a multidimensional integral. We also investigate an alternative to numerical integration which is approximating the underlying probability constraint by Copula (Copula is a simple function, depending on marginal distributions, that models dependence of the joint probability distribution).

By considering, for the moment, chance constraints as regular convex constraints, the problem of interest is symbolized by:

$$\begin{array}{ll} \min & f_0(x,y) \\ \text{s. t.} & f_i(x,y) \leq 0, i \in \mathcal{I}_c \\ & x \in X, y \in Y, \end{array} \tag{1}$$

with f_0, $f_i, i \in \mathcal{I}_c = \{1, \ldots, m_f\}$, convex functions, X a bounded polyhedron and Y a bounded and discrete set. The main difficulty of this problem consists in dealing with Y, the set of integer variables. By relaxing Y to a convex set Y_R (for instance $Y_R = \text{conv}Y$), the following convex problem (this is why problem (1), which is nonconvex, is called a convex MINLP) is gotten:

$$\begin{array}{ll} \min & f_0(x,y) \\ \text{s. t.} & f_i(x,y) \leq 0, i \in \mathcal{I}_c \\ & x \in X, y \in Y_R. \end{array} \tag{2}$$

For example, if $Y = \{0, 1\}$, it leads to take $Y_R = [0, 1]$. This technique is called *relaxing integrality*. The optimal value of (2) is a lower bound for problem (1).

In 1960, in a classic paper, Kelley [42] introduced the cutting plane method for solving convex problems. The strategy was to solve the underlying optimization problem by solving a sequence of linear programs (LPs). At that time, the interest for stochastic optimization and mixed-integer programs started to grow up. In 1962, Benders [5] developed one of the first algorithm to deal with problems having uncertainty parameters and integer variables. But the idea of Benders applies only to problems that involve linear functions.

In 1972, Geoffrion [30] generalized the Bender's approach to a broader class of programs (including MINLP) in which the objective function of those subproblems needs no longer to be linear. Many authors have contributed to the field of MINLP and several algorithms have been developed in the past years.

One of the most famous class of algorithms to solve mixed integer linear problems is the so called *branch-and-bound* algorithms. The first algorithm of this class was developed by Gupta and Ravindran [33] in 1985. The idea of this algorithm consists in solving the relaxed problem (2) first. If all variables are integer, then a solution of problem (1) is obtained. Otherwise, the solution provides a lower bound to the optimal value and in this case a tree search is performed in the space of integer variables. This method is suitable if the cost of solving (2) (and its variants with additional constraints in Y_R and X) is cheaper or if a few of them needs to be solved. If the number of nodes visited in the tree search is too large, then solving the problem becomes a very expensive computational task.

Another important method for solving (1) is the *outer-approximation algorithm* given in [24] and further extended in [28]. The method solves a sequence of nonlinear and mixed linear subproblems, as described below. At iteration k, the method fixes the integer variable y^k and tries to solve, in the continuous variable x, the following subproblem:

$$\begin{array}{ll} \min & f_0(x, y^k) \\ \text{s. t.} & f_i(x, y^k) \leq 0, i \in \mathcal{I}_c \\ & x \in X. \end{array} \tag{3}$$

If this subproblem is feasible, a feasible point to problem (1) is found and, therefore, an upper bound f_{up}^k for its optimal value. On the other hand, if (3) is infeasible the method solves the feasibility subproblem:

$$\begin{array}{ll} \min & u \\ \text{s. t.} & f_i(x, y^k) \leq u, i \in \mathcal{I}_c \\ & x \in X, u \geq 0. \end{array} \tag{4}$$

With a solution of (3) or (4), the linearization of f_0 and f_i can be used to approximate problem (1) by the following MILP:

$$
f_{\text{low}}^k := \begin{cases} \min\limits_{r,x,y} & r \\ \text{ s.t. } & r < f_{\text{up}}^k \\ & f_0(x^j,y^j) + \nabla f_0(x^j,y^j)^\top \begin{bmatrix} x-x^j \\ y-y^j \end{bmatrix} \leq r, \quad j \in T^k \\ & f_i(x^j,y^j) + \nabla f_i(x^j,y^j)^\top \begin{bmatrix} x-x^j \\ y-y^j \end{bmatrix} \leq 0, \quad j \in T^k \cup S^k, \quad i \in \mathcal{I}_c \\ & r \in \mathbb{R},\ x \in X,\ y \in Y\ , \end{cases} \tag{5}
$$

where index sets T^k and S^k are defined as follows

- $T^k := \{j \leq k :$ subproblem (2) was feasible at iteration $j\}$, and
- $S^k := \{j \leq k :$ subproblem (2) was infeasible at iteration $j\}$.

Under convexity of underlying functions, the optimal value f_{low}^k of (5) is a lower bound on the optimal value of (1). Moreover, the $y-$part solution of (5) is the next integer iterate y^{k+1}. The algorithm stops when the difference between upper and lower bounds provided respectively by (3) and (5) is within a given tolerance $\epsilon > 0$. More details about this method will be given in Chapter 3.

The outer approximation algorithm was revisited in 1992 in [56], where the authors proposed a LP/NLP based on the branch and bound strategy in which the explicit solution of a MILP master problem is avoided at each major iteration k. In the context of main interest, the underlying functions might not be differentiable, but subdifferentiable: gradients will be replaced by subgradients. As pointed out in [26], replacing gradients by subgradients in the classic OA algorithm entails a serious issue: the OA algorithm is not convergent if the differentiability assumption is removed. In order to have a convergent OA algorithm for nonsmooth convex MINLP one needs to compute linearizations (cuts) in (5) by using subgradients that satisfy the KKT system of either subproblem (3) or (4), see [26, 75] and Chapter 3 for more details.

Computing solutions and subgradients satisfying the KKT conditions of the nosmooth subproblems is not a trivial task. For instance, the Kelley cutting-plane method and subgradients methods for nosmooth convex optimization problems are ensured to find an optimal solution, but are not ensured to provide a subgradient satisfying the KKT system. Given an optimal solution x^k of (3), there might be infinitely many subgradients of f at x^k if f is nonsmooth. How would a specific subgradient can be chosen in order to satisfy the underlying KKT system? We show in this work that bundle methods give an answer to this crucial question. We will prove that by using a specialized proximal bundle algorithm to solve either (3) or (4), we will be able to compute subgradients that satisfy the KKT conditions and therefore the OA convergence is ensured. The analysis of such proximal bundle algorithm is the first contribution of the present work.

Other important class of algorithms for convex MINLP is based on the Kelley cutting-plane method [76, 77]. These algorithms are able to deal with nonsmooth functions but have, in general, slow convergence. In order to overcome this drawback, [16] and [69] propose regularization techniques to stabilize the iterative process. We follow the lead of [16, 69] and propose to regularize the MILP (5) in order to accelerate OA. We call this resulting method regularized OA. This is the second contribution of this work.

We emphasize that the two first contributions of the present Thesis have been combined in the paper [20], recently published in *Optimization: A Journal of Mathematical Programming and Operations Research*.

As a third contribution, we deal with nonsmooth convex MINLP having some probability constraints. This type of problems appears, for instance, in (stochastic) energy models [3]. In the context of interest, at least one of the constraints in (1) is of the type $f_i(x, y) = \log(p) - \log(P[h(x, y) \geq \xi])$, modeling the chance-constraint $P[h(x, y) \geq \xi] \geq p$, where h is a concave map, $\xi \in \mathbb{R}^m$ is a random vector and $p \in (0, 1)$ is a level parameter. The "log" transform above is very often used to convexify the probability function. As already mentioned, multidimensional numerical integration is employed to evaluate this probability constraint. As an alternative to deal with probability constraints, we will approximate the probability measure P with an appropriate copula. We will investigate suitable copulae to better approximate P in a cheap and easy way.

This work is organized as follows: in Chapter 1 we will elaborate the nonsmooth MINLP and will show a counterexample where the classic OA algorithm loops forever and does not find an optimal solution of (1). In Chapter 2 we will provide and analyze a proximal bundle algorithm that is able to compute both a solution and subgratients satisfying the KKT conditions of the underlying nonsmooth convex optimization problem. As already argued, such bundle algorithm is the working horse in our OA algorithm for dealing with the nonsmooth convex MINLP (1). In Chapter 3, we will provide the regularized OA algorithm. In Chapter 4, we will present the chance constraint MINLP and a family of nonsmooth Copulae which will be used to replace the probability constraint. Finally, in Chapter 5, we will solve a class of hybrid robust and chance-constrained problems that involve a random variable with finite support. In order to evaluate the Copula approach we investigate a power management planning problem described in [3].

Chapter 1

Nonsmooth convex MINLP

The objective of this chapter is to present the deterministic setting of nonsmooth convex mixed-integer nonlinear programs (MINLPs) of the form

$$\begin{aligned} \min_{x,y} \quad & f_0(x,y) \\ \text{s. t.} \quad & f_i(x,y) \leq 0, i \in \mathcal{I}_c \\ & x \in X, y \in Y, \end{aligned} \tag{1.1}$$

where functions $f_0 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}$, $f_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}$ are convex but possibly nonsmooth. The set $X \neq \emptyset$ is a compact polyhedron and $Y \neq \emptyset$ is a compact set of integer variables.

Many real-life optimization problems can be modeled as (1.1). The blackout prevention of electric power system in electrical engineering, for instance, is modeled using MINLP formulation [7]. In chemical engineering, MINLP models are applied in design of water [41]. A recent application in pump scheduling in a class of branched water networks can be found in [11]. Another application is to find the optimal response to a cyber attack [31] in computer science. For more applications refer to [4] and reference therein. In the nondifferentiable setting, MINLP models appear, for instance, in power system optimization [3] and in chance-constrained optimization [17, 69].

Optimization methods have been developed to solve MINLP when the involved functions are differentiable. Some approaches are based on branch-and-bound methods [46] such as [1] and [29]. The work [51] combines branch-price-and-cut strategies with decomposition techniques to provide valid inequalities and strong bounds to guide the search in a branch-and-bound tree. Several others methods deal with these type of problems, see for instance, articles [2, 9, 44, 60].

In the nonsmooth case, an important optimization technique is the extended cutting plane method proposed in [76] and further studied in [26] and [77]. This method is based on the classical cutting plane algorithm given in [42]. Another approach is using bundle methods as proposed by [16], which is a extension of works given in [45] and [68].

In this Thesis, Outer-approximation Algorithms are considered to solve the MINLP. As mentioned before, the OA was introduced in [24] and further extended in [28]. As briefly discussed in the Introduction, OA algorithms solve a sequence of nonlinear subproblems and linear integer subproblems. The choice of this type of approach is because the original problem can be broken in a sequence of easier subproblems and can be faster than other methods.

1.1 An outer-approximation algorithm

A standard outer-approximation algorithm is as following.

Algorithm 1.1. An outer-approximation algorithm

Step 0. *Let* $y^0 \in Y$, $\epsilon > 0$ *and* $tol \geq 0$ *be given. Set* $f_{up}^{-1} = \infty$, $T^{-1} = S^{-1} = \emptyset$, $k = 0$.

Step 1. *Solve either the NLP subproblem (if it is feasible)*

$$\begin{array}{ll} \min\limits_{x} & f_0(x, y^k) \\ s.t. & f_i(x, y^k) \leq 0, i \in \mathcal{I}_c \\ & x \in X, \end{array} \tag{1.2}$$

or the infeasibility subproblem:

$$\begin{array}{ll} \min\limits_{x} & \sum\limits_{i \in \mathcal{I}_c} \max\{f_i(x, y^k), 0\} \\ s.t. & x \in X, \end{array} \tag{1.3}$$

and let the solution be x^k.

Step 2. *Linearize the objective and constraint functions around* (x^k, y^k)*:*

$$f_i(x^k, y^k) + f_i'(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix}, \quad i = 0, \ldots, m_f.$$

Set $T^k = T^{k-1} \cup \{k\}$ *if* x^k *is provided by* (1.2) *or* $S^k = S^{k-1} \cup \{k\}$ *if* x^k *is provided by* (1.3).

Step 3. *If* (1.2) *is feasible and* $f_0(x^k, y^k) < f_{up}^{k-1}$ *then update the current best point by setting* $x^* = x^k, y^* = y^k$ *and* $f_{up}^k = f_0(x^k, y^k)$.

Step 4. *Solve the MILP*

$$\begin{array}{ll} \min & r \\ s.t. & r \leq f_{up}^k - \epsilon \\ & f_0(x^j, y^j) + f_0'(x^j, y^j)^T \begin{bmatrix} x - x^j \\ y - y^j \end{bmatrix} \leq r \ \forall j \in T^k \\ & f_i(x^j, y^j) + f_i'(x^j, y^j)^T \begin{bmatrix} x - x^j \\ y - y^j \end{bmatrix} \leq 0, \forall j \in T^k \cup S^k, i \in \mathcal{I}_c(x^j) \\ & x \in X, y \in Y, r \in \mathbb{R}, \end{array} \tag{1.4}$$

where $\mathcal{I}_c(x^j)$ is the set of active constraint at (x^j, y^j). If (1.4) is infeasible, stop with the $\epsilon-$solution (x^*, y^*). Otherwise, let y^{k+1} be the $y-$part of solution of the above problem, and f^k_{low} its optimal value.

Step 5. *If $f^k_{up} - f^k_{low} < tol$, stop. The pair (x^*, y^*) is a tol-solution of problem (1.1). Otherwise, set $k = k + 1$ and go back to Step 1.*

This algorithm has finite convergence when the involved functions are differentiable and the Slater's assumptions holds [24, 28]. In the next section a nonsmooth MINLP example is considered and is demonstrated that, if arbitrary subgradients to compute cuts (constraints) in problem (1.4) are taken, then the OA algorithm fails to converge.

1.2 The OA algorithm applied to nonsmooth convex MINLP problems: a counterexample

When some (or all) functions f_i, $i = 1, \ldots, m_f$, fail to be continuously differentiable, OA algorithms may cycle indefinitely around non-optimal points. The following example, extracted from [26, § 4.1] illustrates this situation.

Consider the following mixed-integer nonlinear problem:

$$\begin{array}{ll} \min\limits_{x,y} & 2x - y \\ \text{s.t.} & \max\{-\frac{3}{2} - x + y, -\frac{7}{2} + y + x\} \le 0 \\ & -4x + y - 1 \le 0, \\ & 0 \le x \le 2, y \in \{0, 1, 2, 3, 4, 5\}. \end{array} \tag{1.5}$$

Suppose that the initial guess for the integer variable is $y^0 = 3$. So the first nonlinear subproblem becomes

$$\begin{array}{ll} \min\limits_{0 \le x \le 2} & 2x - 3 \\ \text{s. t.} & \max\{\frac{3}{2} - x, -\frac{1}{2} + x\} \le 0 \\ & -4x + 2 \le 0. \end{array} \tag{1.6}$$

Subproblem (1.6) is infeasible. Then the infeasibility subproblem is solved

$$\begin{array}{ll} \min\limits_{0 \le x \le 2} & \max\{\frac{3}{2} - x, -\frac{1}{2} + x, 0\} \\ \text{s. t.} & -4x + 2 \le 0. \end{array} \tag{1.7}$$

By solving (1.7) the solution $x^0 = 1$ with optimal value $\frac{1}{2}$ is obtained. Consequently, $T^0 = \{0\}$ and $S^0 = \{1\}$. To create cuts, it is necessary to have a subgradient at point x^0. Consider the objective function of problem (1.7). The subgradient at the point $x^0 = 1$ is a value $s \in [-1, 1]$ because $\frac{3}{2} - x$ and $-\frac{1}{2} + x$ has the same value at $x^0 = 1$. That means the subgradient at this point is a convex combination between the subgradients of both functions. The subgradient $s = 1$ is chosen yielding the following MILP subproblem

$$\begin{array}{ll} \min\limits_{x,y} & 2x - y \\ \text{s. t.} & x + y - \frac{7}{2} \leq 0 \\ & -4x + y - 1 \leq 0, \\ & 0 \leq x \leq 2, y \in Y = \{0, 1, 2, 3, 4, 5\}. \end{array}$$

The solution of the above problem is $x = \frac{1}{2}$ and $y^1 = 3$. So the OA algorithm repeats the integer variable and enters in an infinite loop. As a result, the OA algorithm fails to solve the nonsmooth problem (1.5).

1.3 The OA algorithm applied to nonsmooth convex MINLP problems: well-chosen subgradients

When the subgradients are carefully chosen the OA algorithm's convergence is guaranteed. In this section the example above is solved with the same initial point $y^0 = 3$. As problem (1.7) is the same, subgradient at point $x^0 = 1$ is a number $s \in [-1, 1]$ as explained before. Now, a subgradient that satisfies the following KKT system of problem (1.7) is chosen:

$$\begin{cases} 0 \in \partial_x[\max\{\dfrac{3}{2} - x^0, -\dfrac{1}{2} + x^0, 0\}] + \bar{\mu}\partial_x[-4x^0 + 2] + N_{[0,2]}(x^0) \\ \bar{\mu}(-4x^0 + 2) = 0 \\ \bar{\mu} \geq 0. \end{cases}$$

As solution x^0 is in the interior of interval $[0, 2]$, the normal cone is the set $\{0\}$. Let $s \in \partial_x[\max\{\dfrac{3}{2} - x^0, -\dfrac{1}{2} + x^0, 0\}]$. The above system is equivalent to

$$\begin{cases} 0 = s - 4\bar{\mu} \\ \bar{\mu}(-4(1) + 2) = 0 \\ \bar{\mu} \geq 0. \end{cases}$$

The unique solution of the above system is $s = \bar{\mu} = 0$. Consequently, $s = 0$ is taken as a subgradient of the objective function in (1.7) at $x^0 = 1$. With this choice the following MILP problem is obtained:

$$\begin{array}{ll} \min\limits_{x,y} & 2x - y \\ \text{s. t.} & y - \frac{5}{2} \leq 0 \\ & -4x + y - 1 \leq 0, \\ & 0 \leq x \leq 2, y \in Y = \{0, 1, 2, 3, 4, 5\}. \end{array}$$

The constraint $y - \frac{5}{2} \leq 0$ eliminates integer values bigger than 2. As a result, only three options remain ($y = 0, 1$ or 2). By solving this subproblem, $\bar{x} = \frac{1}{2}, \bar{y} = 2$ with the optimal value $f^1_{low} = -1$ is gotten. Thus a new integer variable $y^1 = 2$ is known.

Fixing $y^1 = 2$ at problem (1.5) a new nonlinear subproblem is gotten

$$\begin{array}{ll} \min\limits_{x} & 2x - 2 \\ \text{s. t.} & \max\{\frac{1}{2} - x, -\frac{3}{2} + x\} \leq 0 \\ & -4x + 1 \leq 0, \\ & 0 \leq x \leq 2, \end{array}$$

which is feasible and its solution is $x^1 = \frac{1}{2}$ with optimal value $f^1_{up} = -1$. The OA algorithm then stops at Step 5 because $f^1_{up} - f^1_{low} = 0$. Hence, the point $(\bar{x}, \bar{y}) = (\frac{1}{2}, 2)$ is a solution of problem (1.5) with optimal value -1. It turns out that the OA algorithm converges because the chosen subgradient of the nonsmooth constraint satisfies the KKT conditions of problem (1.7).

1.4 General comments

The above example illustrates the fact that subgradients of the involved functions should be carefully chosen in order to ensure convergence of the OA algorithm in the nonsmooth setting. It has been shown in [26] that subgradients of $f_i, i = 0, \dots, m_f$ at x^k must satisfy the KKT system

$$\left\{ \begin{array}{l} 0 \in \partial_x f_0(x^k, y^k) + \sum\limits_{i \in \mathcal{I}_c(x^k)} \bar{\mu}_i^k \partial_x f_i(x^k, y^k) + N_X(x^k) \\ \bar{\mu}_i^k f_i(x^k, y^k) = 0 \,, i \in \mathcal{I}_c(x^k) \\ \bar{\mu}_i^k \geq 0, \, i \in \mathcal{I}_c(x^k) \,, \end{array} \right. \tag{1.8}$$

if problem (1.2) is feasible, or system

$$0 \in \partial_x \left(\sum_{i \in \mathcal{I}_c(x^k)} \max\{f_i(x^k, y^k), 0\} \right) + N_X(x^k), \tag{1.9}$$

otherwise; where $N_X(x^k)$ is the normal cone of X at x^k.

In the paper [26] the authors show how to determine such subgradients for a particular class of functions, in which it is possible to know the vectors $s^i_x \in \partial f(x), i = 1, \dots, p$ yielding

$$\partial f(x) = \left\{ \sum_{i=1}^{p} \alpha_i s^i_x | \sum_{i=1}^{p} \alpha_i = 1, \alpha_i \geq 0, i = 1, \dots, p \right\}.$$

This is, for instance, the case in which every function f_i is the pointwise maximum of finitely many differentiable functions. Hence, the use of the method in [26] is limited because not all the MINLP problems satisfy this assumption. The OA algorithm will be more detailed in Chapter 3. In the next chapter a new algorithm for solving the OA's nonlinear subproblems is proposed. It will be demonstrated that such algorithm provides subgradients satisfying KKT system without further assumptions on problem (1.1). This is a significant improvement on paper [26].

Chapter 2

An exact penalization proximal bundle method

This Chapter corresponds to Section 3 in the work [20], which is a product of this Thesis. Here we are not only concerned with the solution of the OAs subproblems (3) and (4) but also with the calculation of subgradients satisfying the KKT systems (1.8). To accomplish this task we investigate a proximal bundle algorithm.

Bundle methods solve nonsmooth convex optimization problems by requiring only first-order information of the involved functions, and are well-known for their robustness and for having limited memory, that is, one can keep the amount of information (bundle) bounded along the iterative process [10, 39]. The latter is a very useful property when dealing with large scale optimization problems. There are many bundle method variants in the literature: see for example [17, 18, 19, 27, 45]. In this chapter the focus will be on a proximal bundle method variant to solve the nonlinearly constrained nonsmooth optimization subproblems coming from the OA algorithm. It will be presented an algorithm that provides all the required information to ensure convergence of the OA algorithm applied to nonsmmoth convex MINLP problems.

Consider the nonlinear subproblem of OA algorithm for a fixed integer variable y^k:

$$\min_{x\in X} f_0(x,y^k) \;\text{ s.t. }\; f_i(x,y^k) \leq 0\,,\; i \in \mathcal{I}_c := \{1,\ldots,m_f\}\,. \tag{2.1}$$

Problem (2.1) can be feasible or infeasible. If the latter case then the infeasibility problem is solved by

$$\min_{x\in X} \sum_{i\in\mathcal{I}_c} \max\{f_i(x,y^k),0\}\;. \tag{2.2}$$

In both cases, problems (2.1) and (2.2) can be written in a unified and more general manner:

$$\min_{x\in X} \phi(x) \;\text{ s.t. }\; c_i(x) \le 0,\; i \in \mathcal{I}_c\,, \tag{2.3}$$

where with this notation, function $\phi(x)$ is $f_0(x, y^k)$ and $c_i(x)$ is $f_i(x, y^k)$ if subproblem (2.1) is considered or $\phi(x)$ is $\sum_{i\in\mathcal{I}_c} \max\{f_i(x, y^k), 0\}$ otherwise (i.e., (2.2) is considered). In the latter case it does not have nonlinear constraints.

In order to confirm that the set of Lagrange multipliers associated to (2.3) is nonempty and bounded [40], the Slater's condition (if there is at least one nonlinear constraint c_i) is assumed:

H1- There exists $x^0 \in X$ such that $c_i(x^0) < 0$ for all $i \in \mathcal{I}_c$.

As a result, it is ensured the existence of a pair of points $(\bar{x}, \bar{\mu})$ and subgradients $s_\phi \in \partial\phi(\bar{x})$ and $s_i \in \partial c_i(\bar{x})$, $i \in \mathcal{I}_c$, satisfying the following KKT system

$$\begin{cases} -(s_\phi + \sum_{i\in\mathcal{I}_c} \bar{\mu}_i s_i) \in N_X(\bar{x}) \\ c_i(\bar{x}) \le 0 \le \bar{\mu}_i,\; i \in \mathcal{I}_c \\ \bar{x} \in X,\; \bar{\mu}_i c_i(\bar{x}) = 0,\; i \in \mathcal{I}_c\,. \end{cases} \tag{2.4}$$

2.1 Description of the method

The method generates a sequence of feasible iterates $\{x^\ell\} \subset X$. For each point x^ℓ, an oracle is called to compute $\phi(x^\ell)$, $c_i(x^\ell)$, $i \in \mathcal{I}_c$ and subgradients $s_\phi^\ell \in \partial\phi(x^\ell)$, $s_i^\ell \in \partial c_i(x^\ell)$, $i \in \mathcal{I}_c$. With such information, the method creates cutting-plane models for the functions

$$\begin{aligned} \check{\phi}^\ell(x) &:= \max_{j\in\mathcal{B}^\ell}\{\phi(x^j) + \langle s_\phi^j, x - x^j\rangle\} &&\le \phi(x), \\ \check{c}_i^\ell(x) &:= \max_{j\in\mathcal{B}_i^\ell}\{c_i(x^j) + \langle s_i^j, x - x^j\rangle\} &&\le c_i(x) \quad i \in \mathcal{I}_c\ . \end{aligned} \tag{2.5}$$

The index sets $\mathcal{B}^\ell$, $\mathcal{B}_i^\ell$ are in general subsets of $\{1, \ldots, \ell\}$, but can also contain some index of an artificial/aggregate linearization.

Given a stability center $\hat{x}^\ell \in X$ and a prox-parameter $\tau^\ell > 0$, the new iterate $x^{\ell+1}$ is obtained by solving the QP subproblem

$$\min_{x\in X} \check{\phi}^\ell(x) + \frac{1}{2\tau^\ell}\|x - \hat{x}^\ell\|^2 \quad \text{s.t.} \quad \check{c}_i^\ell(x) \le 0, \quad i \in \mathcal{I}_c\,, \tag{2.6}$$

that can be rewritten as

$$\begin{cases} \min\limits_{(r,x)\in\mathbb{R}\times X} & r + \frac{1}{2\tau^\ell}\|x - \hat{x}^\ell\|^2 \\ \text{s.t.} & \phi(x^j) + \langle s_\phi^j, x - x^j\rangle \le r\,, \quad j \in \mathcal{B}^\ell \\ & c_i(x^j) + \langle s_i^j, x - x^j\rangle \le 0\,, \quad j \in \mathcal{B}_i^\ell,\, i \in \mathcal{I}_c\,. \end{cases} \tag{2.7}$$

The stability center $\hat{x}^\ell$ is some previous iterates, usually is the best point generated by the algorithm. In order to define what is the "best point so far", a nonsmooth penalization function and a penalization parameter $\rho > 0$ will be employed:

$$f_\rho(x) := \phi(x) + \rho \sum_{i\in\mathcal{I}_c} \max\{c_i(x), 0\} \quad \text{and} \quad \check{f}_\rho^\ell(x) := \check{\phi}^\ell(x) + \rho \sum_{i\in\mathcal{I}_c} \max\{\check{c}_i^\ell(x), 0\}. \tag{2.8}$$

A classification rule decides when to update $\hat{x}^\ell$: if

$$(0 \le) \quad \kappa\,(f_\rho(\hat{x}^\ell) - \check{f}_\rho^\ell(x^{\ell+1})) \le f_\rho(\hat{x}^\ell) - f_\rho(x^{\ell+1}), \quad \text{with } \kappa \in (0,1) \tag{2.9}$$

then $\hat{x}^{\ell+1} := x^{\ell+1}$, otherwise $\hat{x}^{\ell+1} := \hat{x}^\ell$. In other words, the stability center is updated only when the new candidate provides enough decrease with respect to the penalization function: at least a fraction of the decrease provided by the model: $\kappa\,(f_\rho(\hat{x}^\ell) - \check{f}_\rho^\ell(x^{\ell+1}))$. The following is a useful result for the remaining of this chapter.

Proposition 2.1. *Consider problem* (2.3) *and assume the involved functions to be convex and $X \neq \emptyset$ a polyhedron.*

a) The vector $x^{\ell+1}$ solves (2.6) *if and only if $x^{\ell+1} \in X$, $\check{c}_i^\ell(x^{\ell+1}) \le 0$, $i \in \mathcal{I}_c$, and there exist vectors $\hat{s}_\phi^\ell \in \partial\check{\phi}^\ell(x^{\ell+1})$, $\hat{s}_i^\ell \in \partial\check{c}_i^\ell(x^{\ell+1})$, $i \in \mathcal{I}_c$, $s_X^\ell \in N_X(x^{\ell+1})$ and stepsizes $\mu_i^\ell \ge 0$ such that $\mu_i^\ell\check{c}_i^\ell(x^{\ell+1}) = 0$ and*

$$x^{\ell+1} = \hat{x}^\ell - \tau^\ell d^\ell, \quad d^\ell := \hat{s}_\phi^\ell + \sum_{i\in\mathcal{I}_c} \mu_i^\ell \hat{s}_i^\ell + s_X^\ell\,. \tag{2.10}$$

b) Let $\alpha^j \ge 0$ (resp. $\lambda_i^j \ge 0$) be Lagrange multiplier associated with the constraint $\phi(x^j) + \langle s_\phi^j, x - x^j\rangle \le r$ (resp. $c_i(x^j) + \langle s_i^j, x - x^j\rangle \le 0$) in (2.7)*. Then,*

$$\sum_{j\in\mathcal{B}^\ell} \alpha^j = 1, \quad \mu_i^\ell = \sum_{j\in\mathcal{B}_i^\ell} \lambda_i^j, \quad \hat{s}_\phi^\ell = \sum_{j\in\mathcal{B}^\ell} \alpha^j s_\phi^j \quad \textit{and} \quad \hat{s}_i^\ell = \frac{1}{\mu_i^\ell} \sum_{j\in\mathcal{B}_i^\ell} \lambda_i^j s_i^j\,.$$

c) The aggregate linearizations $\phi^{-\ell}(x) := \check{\phi}^\ell(x^{\ell+1}) + \langle \hat{s}_\phi^\ell, x - x^{\ell+1}\rangle$ and $c_i^{-\ell}(x) := \check{c}_i^\ell(x^{\ell+1}) + \langle \hat{s}_i^\ell, x - x^{\ell+1}\rangle$ satisfies $\phi^{-\ell}(x) \le \phi(x)$ and $c_i^{-\ell}(x) \le c_i(x)$ for all $x \in \mathbb{R}^{n_x}$.

Let $f_\rho^{-\ell}(x) := \check{f}_\rho^\ell(x^{\ell+1}) + \langle d^\ell, x - x^{\ell+1}\rangle$ be the aggregate linearization of f_ρ. If $\rho \geq \max_{i\in\mathcal{I}_c} \mu_i^\ell$ then $f_\rho^{-\ell}(x) \leq f_\rho(x) + i_X(x)$ is obtained for all $x \in \mathbb{R}^{n_x}$, where i_X is the indicator function of the polyhedral set X, i.e., $i_X(x) = 0$ if $x \in X$ and $i_X(x) = \infty$ otherwise.

d) Let $\hat{e}_\phi^\ell := \phi(\hat{x}^\ell) - \phi^{-\ell}(\hat{x}^\ell)$ and $\hat{e}_i^\ell := c_i(\hat{x}^\ell) - c_i^{-\ell}(\hat{x}^\ell)$. Then

$$\hat{e}_\phi^\ell,\, \hat{e}_i^\ell \geq 0, \quad \hat{s}_\phi^\ell \in \partial_{\hat{e}_\phi^\ell}\phi(\hat{x}^\ell), \quad \textit{and} \quad \hat{s}_i^\ell \in \partial_{\hat{e}_i^\ell} c_i(\hat{x}^\ell)\,.$$

Let $\hat{e}^\ell := f_\rho(\hat{x}^\ell) + i_X(\hat{x}^\ell) - f_\rho^{-\ell}(\hat{x}^\ell)$ be the aggregate error. If $\rho \geq \max_{i\in\mathcal{I}_c} \mu_i^\ell$ then

$$\hat{e}^\ell \geq 0 \quad d^\ell \in \partial_{\hat{e}^\ell}[f_\rho(\hat{x}^\ell) + i_X(\hat{x}^\ell)], \quad \textit{and} \quad \hat{e}^\ell \geq \hat{e}_\phi^\ell + \sum_{i\in\mathcal{I}_c} \mu_i^\ell\, \hat{e}_i^\ell\,.$$

e) Suppose that $\lim_{\ell\to\infty} \hat{e}^\ell = 0$, and let $\hat{x}$ be a cluster point of $\{\hat{x}^\ell\}$, i.e., $\lim_{\ell\in K} \hat{x}^\ell = \hat{x}$. Let also $\mu_i \geq 0$ be a cluster point of $\{\mu_i^\ell\}_K$, $i \in \mathcal{I}_c$. Then any cluster points $d, s_\phi, s_i \in \mathbb{R}^{n_x}$ of $\{d^\ell\}_K$, $\{s_\phi^\ell\}_K$ and $\{s_i^\ell\}_K$ satisfy

$$\hat{s}_\phi \in \partial\phi(\hat{x}) \quad \textit{and} \quad \hat{s}_i \in \partial c_i(\hat{x}) \quad \textit{if } \mu_i > 0\,.$$

If $\rho \geq \max_{i\in\mathcal{I}_c} \mu_i^\ell$, then $d \in \partial[f_\rho(\hat{x}) + i_X(\hat{x})]$.

Proof:

a) It follows from [57, p.215] that $\partial i_X(x) = N_X(x)$ for $x \in X$. Problem (2.6) can be written as

$$\min_{x\in\mathbb{R}^{n_x}} \check{\phi}^\ell(x) + \frac{1}{2\tau^\ell}\|x - \hat{x}^\ell\|^2 + i_X(x) \quad \text{s.t.} \quad \check{c}_i^\ell(x) \leq 0, \quad i \in \mathcal{I}_c\,.$$

From KKT's conditions the system holds:

$$\begin{cases} \hat{s}_\phi^\ell + \frac{1}{\tau^\ell}(x^{\ell+1} - \hat{x}^\ell) + s_X^\ell + \sum_{i\in\mathcal{I}_c} \mu_i^\ell \hat{s}_i^\ell = 0, \\ \mu_i^\ell \geq 0, i \in \mathcal{I}_c, \\ \mu_i^\ell \check{c}_i^\ell(x^{\ell+1}) = 0, i \in \mathcal{I}_c. \end{cases}$$

Putting all subgradients together the results are acquired.

b) The Lagrangian function of problem (2.7) for $x \in X$ and $\tilde{\alpha}, \tilde{\lambda} \geq 0$, is

$$\begin{aligned} L(x, r; \tilde{\alpha}, \tilde{\lambda}) :=& r + \frac{1}{2\tau^\ell}\|x - \hat{x}^\ell\|^2 + \sum_{j\in\mathcal{B}^\ell} \tilde{\alpha}^j(\phi(x^j) + \langle s_\phi^j, x - x^j\rangle - r) \\ &+ \sum_{i\in\mathcal{I}_c}\sum_{j\in\mathcal{B}_i^\ell} \tilde{\lambda}_i^j(c_i(x^j) + \langle s_i^j, x - x^j\rangle)\,. \end{aligned} \tag{2.11}$$

Notice that the minimum of L over $(r,x) \in \mathbb{R} \times X$ is well defined if and only if $\sum_{j\in\mathcal{B}^\ell} \tilde{\alpha}^j = 1$ (otherwise $\inf_{(r,x)\in\mathbb{R}\times X} L(x,r;\tilde{\alpha},\tilde{\lambda}) = -\infty$). Let $(r^{\ell+1}, x^{\ell+1})$ be the solution of (2.7) and (α, λ) be the associate Langrange multipliers. It follows from the KKT conditions that $\alpha^j > 0$ implies

$$\phi(x^j) + \langle s_\phi^j, x^{\ell+1} - x^j\rangle = r^{\ell+1} = \check{\phi}^\ell(x^{\ell+1}), \tag{2.12}$$

and $\lambda_i^j > 0$ implies

$$c_i(x^j) + \langle s_i^j, x^{\ell+1} - x^j\rangle = 0 = \check{c}_i^\ell(x^{\ell+1}). \tag{2.13}$$

Differentiating both sides of (2.12) and (2.13) in relation to $x^{\ell+1}$ the result is $s_\phi^j \in \partial\check{\phi}^\ell(x^{\ell+1})$ and $s_i^j \in \partial\check{c}_i^\ell(x^{\ell+1})$. The subdifferential of a convex function is convex, then

$$\sum_{j\in\mathcal{B}^\ell} \alpha^j s_\phi^j \in \partial\check{\phi}^\ell(x^{\ell+1})$$

and

$$\frac{1}{\sum_{j\in\mathcal{B}_i^\ell} \lambda_i^j} \sum_{j\in\mathcal{B}_i^\ell} \lambda_i^j s_i^j \in \partial\check{c}_i^\ell(x^{\ell+1}).$$

Moreover, it follows from convexity of subproblem (2.7) and uniqueness of its solution that $(r^{\ell+1}, x^{\ell+1})$ is also the unique solution of

$$\min_{(r,x)\in\mathbb{R}\times X} L(x,r;\alpha,\lambda)$$

which is equivalent (in terms of solution x) to

$$\min_{x\in X} \frac{1}{2\tau^\ell}\|x - \hat{x}^\ell\|^2 + \langle \sum_{j\in\mathcal{B}^\ell} \alpha^j s_\phi^j, x\rangle + \sum_{i\in\mathcal{I}_c}\langle \sum_{j\in\mathcal{B}_i^\ell} \lambda_i^j s_i^j, x\rangle\,. \tag{2.14}$$

The KKT conditions for this problem lead to

$$\frac{1}{\tau^\ell}(x^{\ell+1} - \hat{x}^\ell) + \sum_{j\in\mathcal{B}^\ell} \alpha^j s_\phi^j + \sum_{i\in\mathcal{I}_c}\sum_{j\in\mathcal{B}_i^\ell} \lambda_i^j s_i^j + s_X^\ell = 0$$

which is equivalent to

$$-\left(\frac{x^{\ell+1} - \hat{x}^\ell}{\tau^\ell} + \sum_{j\in\mathcal{B}^\ell} \alpha^j s_\phi^j + \sum_{i\in\mathcal{I}_c}\sum_{j\in\mathcal{B}_i^\ell} \lambda_i^j s_i^j\right) \in N_X(x^{\ell+1})\,.$$

By defining μ_i^ℓ, $\hat{s}_\phi^\ell$ and $\hat{s}_i^\ell$ as in item b) the above relation is the same as (2.10).

c) By item a), $\hat{s}_\phi^\ell \in \partial\check{\phi}^\ell(x^{\ell+1})$. So

$$\phi^{-\ell}(x) = \check{\phi}^\ell(x^{\ell+1}) + \langle \hat{s}_\phi^\ell, x - x^{\ell+1}\rangle \leq \check{\phi}^\ell(x) \leq \phi(x) \;\forall x \in \mathbb{R}^{n_x}.$$

The similar way implies

$$c_i^{-\ell}(x) = \check{c}_i^\ell(x^{\ell+1}) + \langle \hat{s}_i^\ell, x - x^{\ell+1}\rangle \leq \check{c}_i^\ell(x) \leq c_i(x) \;\forall x \in \mathbb{R}^{n_x}.$$

The last result follows below

$$\begin{aligned}
f_\rho^{-\ell}(x) &= \check{f}_\rho^\ell(x^{\ell+1}) + \langle d^\ell, x - x^{\ell+1}\rangle \\
&= \check{\phi}^\ell(x^{\ell+1}) + \rho\sum_{i\in\mathcal{I}_c}\max\{\check{c}_i^\ell(x^{\ell+1}), 0\} + \langle d^\ell, x - x^{\ell+1}\rangle \\
&= \check{\phi}^\ell(x^{\ell+1}) + \rho\sum_{i\in\mathcal{I}_c}\max\{\check{c}_i^\ell(x^{\ell+1}), 0\} + \langle \hat{s}_\phi^\ell + \sum_{i\in\mathcal{I}_c}\mu_i^\ell\hat{s}_i^\ell + s_X^\ell, x - x^{\ell+1}\rangle \\
&\leq \check{\phi}^\ell(x) + i_X(x^{\ell+1}) + \langle s_X^\ell, x - x^{\ell+1}\rangle + \rho\sum_{i\in\mathcal{I}_c}\max\{\check{c}_i^\ell(x^{\ell+1}), 0\} \\
&\qquad\qquad + \langle \textstyle\sum_{i\in\mathcal{I}_c}\mu_i^\ell\hat{s}_i^\ell, x - x^{\ell+1}\rangle.
\end{aligned}$$

The last inequality follows from the aggregate linearizations $\phi^{-\ell}(x) \leq \phi(x)$ which was proved above. The next inequality follows from subgradient of i_X at point $x^{\ell+1}$.

$$\begin{aligned}
f_\rho^{-\ell}(x) &\leq \phi(x) + i_X(x) + \sum_{i\in\mathcal{I}_c}\left[\rho\max\{\check{c}_i^\ell(x^{\ell+1}), 0\} + \langle \mu_i^\ell\hat{s}_i^\ell, x - x^{\ell+1}\rangle\right] \\
&= \phi(x) + i_X(x) + \sum_{i\in\mathcal{I}_c}\langle \mu_i^\ell\hat{s}_i^\ell, x - x^{\ell+1}\rangle \\
&= \phi(x) + i_X(x) + \sum_{i\in\mathcal{I}_c}\mu_i^\ell[\check{c}_i^\ell(x^{\ell+1}) + \langle \hat{s}_i^\ell, x - x^{\ell+1}\rangle] \\
&\leq \phi(x) + i_X(x) + \sum_{i\in\mathcal{I}_c}\mu_i^\ell\check{c}_i^\ell(x) \\
&\leq \phi(x) + i_X(x) + \sum_{i\in\mathcal{I}_c}\mu_i^\ell c_i(x) \\
&\leq \phi(x) + i_X(x) + \sum_{i\in\mathcal{I}_c}\mu_i^\ell\max\{c_i(x), 0\} \\
&\leq \phi(x) + i_X(x) + \sum_{i\in\mathcal{I}_c}\rho\max\{c_i(x), 0\} \\
&= f_\rho(x) + i_X(x).
\end{aligned}$$

In the development above was used that $\check{c}_i^\ell(x^{\ell+1}) = 0$ from (2.13), $c_i^{-\ell}(x) \leq c_i(x)$ and the assumption $\rho \geq \max\limits_{i\in\mathcal{I}_c}\mu_i^\ell$.

d) It follows from item c) that $\phi \geq \phi^{-\ell}$ and $c_i \geq c_i^{-\ell}$. Then the aggregate error $\hat{e}_\phi^\ell$

and $\hat{e}_i^\ell$ are both nonnegative. Moreover,

$$\begin{aligned}\phi(x) \geq \phi^{-\ell}(x) &= \check{\phi}^\ell(x^{\ell+1}) + \langle \hat{s}_\phi^\ell, x - x^{\ell+1}\rangle \\ &= \phi(\hat{x}^\ell) + (-\phi(\hat{x}^\ell) + \check{\phi}^\ell(x^{\ell+1}) + \langle \hat{s}_\phi^\ell, \hat{x}^\ell - x^{\ell+1}\rangle) + \langle \hat{s}_\phi^\ell, x - \hat{x}^\ell\rangle \\ &= \phi(\hat{x}^\ell) - \hat{e}_\phi^\ell + \langle \hat{s}_\phi^\ell, x - \hat{x}^\ell\rangle,\end{aligned}$$

showing that $\hat{s}_\phi^\ell \in \partial_{\hat{e}_\phi^\ell}\phi(\hat{x}^\ell)$. The proof that $\hat{s}_i^\ell \in \partial_{\hat{e}_i^\ell} c_i(\hat{x}^\ell)$ is analogous. Next step is to demonstrate that $\hat{e}^\ell \geq 0$. Consider the following development:

$$\begin{aligned}\hat{e}^\ell &= f_\rho(\hat{x}^\ell) - (\check{f}_\rho(x^{\ell+1}) + \langle d^\ell, \hat{x}^\ell - x^{\ell+1}\rangle) \\ &= \phi(\hat{x}^\ell) + \rho\sum_{i\in\mathcal{I}_c}\max\{c_i(\hat{x}^\ell), 0\} - \Bigg(\check{\phi}^\ell(x^{\ell+1}) + \rho\sum_{i\in\mathcal{I}_c}\max\{\check{c}_i^\ell(x^{\ell+1}), 0\} \\ &\qquad\qquad + \langle \hat{s}_\phi^\ell + \sum_{i\in\mathcal{I}_c}\mu_i^\ell\hat{s}_i^\ell + s_X^\ell, \hat{x}^\ell - x^{\ell+1}\rangle\Bigg) \\ &= \phi(\hat{x}^\ell) - (\check{\phi}^\ell(x^{\ell+1}) + \langle \hat{s}_\phi^\ell, \hat{x}^\ell - x^{\ell+1}\rangle) - \langle s_X^\ell, \hat{x}^\ell - x^{\ell+1}\rangle) \\ &\qquad + \sum_{i\in\mathcal{I}_c}\left[\rho\max\{c_i(\hat{x}^\ell), 0\} - \left(\rho\max\{\check{c}_i^\ell(x^{\ell+1}), 0\} + \langle \mu_i^\ell\hat{s}_i^\ell, \hat{x}^\ell - x^{\ell+1}\rangle\right)\right] \\ &\geq \hat{e}_\phi^\ell + \sum_{i\in\mathcal{I}_c}\left[\rho\max\{c_i(\hat{x}^\ell), 0\} - \left(\rho\max\{\check{c}_i^\ell(x^{\ell+1}), 0\} + \langle \mu_i^\ell\hat{s}_i^\ell, \hat{x}^\ell - x^{\ell+1}\rangle\right)\right],\end{aligned}$$

where the last inequality is because $s_X^\ell \in \partial N_X(x^{\ell+1})$. Notice that $\check{c}_i^\ell(x^{\ell+1}) \leq 0$ and $\mu_i^\ell\,\check{c}_i^\ell(x^{\ell+1}) = 0$ by the KKT conditions (item a). Therefore, $\max\{\check{c}_i^\ell(x^{\ell+1}), 0\} = 0$ and

$$\begin{aligned}\hat{e}^\ell &\geq \hat{e}_\phi^\ell + \sum_{i\in\mathcal{I}_c}\left[\rho\max\{c_i(\hat{x}^\ell), 0\} - \langle \mu_i^\ell\hat{s}_i^\ell, \hat{x}^\ell - x^{\ell+1}\rangle\right] \\ &\geq \hat{e}_\phi^\ell + \sum_{i\in\mathcal{I}_c}\left[\mu_i^\ell\max\{c_i(\hat{x}^\ell), 0\} - \langle \mu_i^\ell\hat{s}_i^\ell, \hat{x}^\ell - x^{\ell+1}\rangle\right] \\ &\geq \hat{e}_\phi^\ell + \sum_{i\in\mathcal{I}_c}\left[\mu_i^\ell c_i(\hat{x}^\ell) - (\mu_i^\ell\check{c}_i^\ell(x^{\ell+1}) + \langle \mu_i^\ell\hat{s}_i^\ell, \hat{x}^\ell - x^{\ell+1}\rangle)\right] \\ &= \hat{e}_\phi^\ell + \sum_{i\in\mathcal{I}_c}\mu_i^\ell\hat{e}_i^\ell \geq 0\,,\end{aligned}$$

where the second inequality above is due to the assumption that $\rho \geq \max_{i\in\mathcal{I}_c}\mu_i^\ell$. Under this assumption, then item c) ensures $f_\rho(x) + i_X(x) \geq f_\rho^{-\ell}(x)$ for all $x \in \mathbb{R}^{n_x}$. It is remaining to show that $d^\ell \in \partial_{\hat{e}^\ell}[f_\rho(\hat{x}^\ell) + i_X(\hat{x}^\ell)]$. It follows from item c) that

$$\begin{aligned}f_\rho(x) + i_X(x) \geq f_\rho^{-\ell}(x) &= \check{f}_\rho^\ell(x^{\ell+1}) + \langle d^\ell, x - x^{\ell+1}\rangle \\ &= f_\rho(\hat{x}^\ell) + (-f_\rho(\hat{x}^\ell) + \check{f}_\rho^\ell(x^{\ell+1}) + \langle d^\ell, \hat{x}^\ell - x^{\ell+1}\rangle) \\ &\qquad + \langle d^\ell, x - \hat{x}^\ell\rangle \\ &= f_\rho(\hat{x}^\ell) - \hat{e}^\ell + \langle d^\ell, x - \hat{x}^\ell\rangle \\ &= f_\rho(\hat{x}^\ell) + i_X(\hat{x}^\ell) + \langle d^\ell, x - \hat{x}^\ell\rangle - \hat{e}^\ell\,.\end{aligned}$$

Thus item d) has been proved.

e) If $\rho > \max_{i\in\mathcal{I}_c} \mu_i^\ell$, then item d) ensures that

$$f_\rho(x) + i_X(x) \geq f_\rho(\hat{x}^\ell) + i_X(\hat{x}^\ell) + \langle d^\ell, x - \hat{x}^\ell\rangle - \hat{e}^\ell \quad \forall\, x \in \mathbb{R}^{n_x}\,.$$

Take the limit with $\ell \in K$ in the above inequality to conclude that $d \in \partial[f_\rho(\hat{x}) + i_X(\hat{x})]$ (recall that f_ρ is continuous). The remaining results follow from the same reasoning and inequality $\hat{e}^\ell \geq \hat{e}^\ell_\phi + \sum_{i\in\mathcal{I}_c} \mu_i^\ell \hat{e}_i^\ell$ (≥ 0).

■

It follows from standard results on exact penalization of constrained optimization problems (see for example Theorem 6.9 in [59]) that if $\rho > \max_{i\in\mathcal{I}_c} \mu_i^\ell$, and a Slater point for (2.3) exists, then the point $x^{\ell+1}$ solution of (2.6) also solves the QP

$$\min_{x\in X} \check{f}^\ell_\rho(x) + \frac{1}{2\tau^\ell}\|x - \hat{x}^\ell\|^2\,. \tag{2.15}$$

This argument is employed in the proof of Theorem 2.1 below.

Bundle method algorithm developed in this work is presented in the sequence.

Algorithm 2.1. An exact penalization proximal bundle algorithm

Step 0. *(Initialization) Select* $\kappa \in (0,1)$, $\tau_{\max} \geq \tau^1 \geq \tau_{\min} > 0$ *and a penalization parameter* $\rho > 0$. *Choose* $x^1 \in X$ *and stopping tolerances* ϵ_0, ϵ_1, $\epsilon_2 > 0$. *Call the oracle to compute* $(\phi(x^1), s^1_\phi)$ *and* $(c_i(x^1), s^1_i)$, $i \in \mathcal{I}_c$. *Set* $\hat{x}^1 \leftarrow x^1$, $\ell \leftarrow 1$, $\hat{\ell} \leftarrow 1$, $\mathcal{B}^1 \leftarrow \{1\}$.

Step 1. *(Next iterate) Obtain* $x^{\ell+1}$ *by solving* (2.7). *Let* μ_i^ℓ *as in Proposition 2.1 b). If* $\rho \leq \mu^{\max} := \max_{i\in\mathcal{I}_c} \mu_i^\ell$, *define* $\rho \leftarrow \mu^{\max} + 1$. *Set* $d^\ell \leftarrow (\hat{x}^\ell - x^{\ell+1})/\tau^\ell$, *and* $\hat{e}^\ell \leftarrow f_\rho(\hat{x}^\ell) - \check{f}^\ell_\rho(x^{\ell+1}) - \tau^\ell\|d^\ell\|^2$. *Compute (approximated) subgradients* $\hat{s}^\ell_\phi$ *and* $\hat{s}^\ell_i$ *for* $i \in \mathcal{I}_c$ *s.t.* $\mu_i^\ell > 0$ *as described in Proposition 2.1 b).*

Step 2. *(Stopping test) If* $\max_{i\in\mathcal{I}_c} c(\hat{x}^\ell) \leq \epsilon_0$, $\hat{e}^\ell \leq \epsilon_1$ *and* $\|\hat{d}^\ell\| \leq \epsilon_2$, *stop. Return* $\hat{x}^\ell$, $(\phi(\hat{x}^\ell), \hat{s}^\ell_\phi)$ *and* $(c_i(\hat{x}^\ell), \hat{s}^\ell_i)$ *for* $i \in \mathcal{I}_c$ *such that* $\mu_i^\ell > 0$.

Step 3. *(Oracle call) Call the oracle to compute* $(\phi(x^{\ell+1}), s^{\ell+1}_\phi)$ *and* $(c_i(x^{\ell+1}), s^{\ell+1}_i)$, $i \in \mathcal{I}_c$.

Step 4. *(Descent test). If* (2.9) *holds, then set* $\hat{x}^{\ell+1} \leftarrow x^{\ell+1}$, $\hat{\ell} \leftarrow \ell + 1$ *and choose* $\tau^{\ell+1} \in [\tau^\ell, \tau_{\max}]$; *otherwise set* $\hat{x}^{\ell+1} \leftarrow \hat{x}^\ell$ *and choose* $\tau^{\ell+1} \in [\tau_{\min}, \tau^\ell]$.

Step 5 *(Bundle management) Choose* $\mathcal{B}^{\ell+1} \supset \{\ell+1, \hat{\ell}, -\ell\}$, $\mathcal{B}_i^{\ell+1} \supset \{\ell+1, \hat{\ell}, -\ell\}$, $i \in \mathcal{I}_c$. *Set* $\ell \leftarrow \ell + 1$ *and go back to Step 1.*

The penalization parameter ρ is only used in Steps 1 and 4, and it is not considered in the QP subproblem. Therefore, the algorithm is not hindered by potentially large values of ρ. This is the main advantage of this algorithm over the ones proposed in [43] that employs the penalization parameter in the objective function of the QP subproblem.

It is worth mentioning that the ingredients d^ℓ and $\hat{e}^\ell$ are easily computed in Step 1 of the algorithm and coincide with their definitions given in Propositions 2.1 a) and 2.1 d), respectively.

The rule given in Step 5 above is a very economical one, since the information bundle can be reduced to m triples of linearizations: the one issued by the new oracle information ($\ell + 1$), the one given by the last descent iterate ($\hat{\ell}$) and the artificial/aggregated linearization represented by $-\ell$. Other bundle methods in the literature (e.g. [18]) do not require keeping in the information bundle the linearization related to the last descent iterate. However, having $\hat{\ell} \in \mathcal{B}^{\ell+1}$ in all iterations ℓ facilitates the mathematical proof that the algorithm provides subgradients satisfying the KKT system (2.4) (see Theorem 2.1 below).

2.2 Convergence analysis

In what follows it is demonstrated that Algorithm 2.1 converges to a solution of (2.3) and, moreover, provides subgradients and multipliers satisfying the KKT system (2.4). First, it is shown that the penalization parameter ρ (only used to update the stability center and to compute the aggregate error in Step 1 of the algorithm) is bounded. To this end, it is necessary to prove that multipliers associated with the QP subproblem (2.6) are bounded as well.

Proposition 2.2. *Suppose that X is a bounded set and there exists $x^0 \in X$ such that $c_i(x^0) < 0$ for all $i \in \mathcal{I}_c$. If $\tau^\ell \geq \tau_{\min} > 0$, then the sequence of Lagrange multipliers $\{\mu_i^\ell\}$ of* (2.6) *are bounded.*

Proof: It follows from Proposition 2.1 items a) and b) that

$$d^\ell = \hat{s}_\phi^\ell + \sum_{i \in \mathcal{I}_c} \mu_i^\ell \hat{s}_i^\ell + s_X^\ell = \hat{s}_\phi^\ell + \sum_{i \in \mathcal{I}_c} \sum_{j \in \mathcal{B}_i^\ell} \lambda_i^j s_i^j + s_X^\ell,$$

where $\lambda_i^j \geq 0$ is the Lagrange multiplier associated with the constraint $c_i(x^j) + \langle s_i^j, x - x^j \rangle \leq 0$ in (2.7). By defining

$$\bar{\mu}^\ell = \sum_{i \in \mathcal{I}_c} \mu_i^\ell = \sum_{i \in \mathcal{I}_c} \sum_{j \in \mathcal{B}_i^\ell} \lambda_i^j \quad \text{and} \quad \bar{s}^\ell = \frac{1}{\bar{\mu}^\ell} \sum_{i \in \mathcal{I}_c} \mu_i^\ell \hat{s}_i^\ell = \frac{1}{\bar{\mu}^\ell} \sum_{i \in \mathcal{I}_c} \sum_{j \in \mathcal{B}_i^\ell} \lambda_i^j s_i^j$$

$d^\ell = \hat{s}_\phi^\ell + \bar{\mu}^\ell \bar{s}^\ell + s_X^\ell$ is obtained. Notice that, by proving that $\{\bar{\mu}^\ell\}$ is a bounded sequence, it is also proven that every sequence $\{\mu_i^\ell\} = \{\sum_{j\in\mathcal{B}_i^\ell} \lambda_i^j\}$ is bounded (because $\lambda_i^j \geq 0$, and therefore $\mu_i^\ell \geq 0$). To this end, assuming without loss of generality that $\bar{\mu}^\ell > 0$ to define $p^\ell = \bar{s}^\ell + \frac{1}{\bar{\mu}^\ell} s_X^\ell$. Since $x^{\ell+1} = \hat{x}^\ell - \tau^\ell d^\ell$ by (2.10), it is conclude that

$$\bar{\mu}^\ell p^\ell = \bar{\mu}^\ell \bar{s}^\ell + s_X^\ell = d^\ell - \hat{s}_\phi^\ell = -\left(\frac{x^{\ell+1} - \hat{x}^\ell}{\tau^\ell} + \hat{s}_\phi^\ell\right).$$

As a result,

$$\bar{\mu}^\ell \|p^\ell\|^2 = \langle p^\ell, -\left(\frac{x^{\ell+1} - \hat{x}^\ell}{\tau^\ell} + \hat{s}_\phi^\ell\right)\rangle \leq \|p^\ell\| \left(\frac{\|x^{\ell+1} - \hat{x}^\ell\|}{\tau^\ell} + \|\hat{s}_\phi^\ell\|\right).$$

As, $\tau^\ell \geq \tau_{\min} > 0$, X is a compact set and ϕ is a convex function, the subdifferential of ϕ is compact on X and therefore there exists a constant $M > 0$ bounding $\|x^{\ell+1} - \hat{x}^\ell\|/\tau^\ell + \|\hat{s}_\phi^\ell\|$. Thus,

$$\bar{\mu}^\ell \leq \frac{\|p^\ell\|}{\|p^\ell\|^2} M = \frac{1}{\|p^\ell\|} M \quad \forall\, \ell.$$

By showing that $\|p^\ell\|$ is bounded away from zero it is also proven that $\{\bar{\mu}^\ell\}$ is bounded from above. Let $\gamma = \max_{i\in\mathcal{I}_c} c_i(x^0)$. The definitions of $\hat{s}_i^\ell$ and s_X^ℓ implies that

$$\check{c}_i^\ell(x^{\ell+1}) + \langle \hat{s}_i^\ell, x^0 - x^{\ell+1}\rangle \leq \check{c}_i^\ell(x^0) \leq c_i(x^0) \leq \gamma < 0 \quad \text{and} \quad \langle s_X^\ell, x^0 - x^{\ell+1}\rangle \leq 0.$$

By multiplying the first relation above by $\mu_i^\ell > 0$ and remembering that $\mu_i^\ell\, \check{c}_i^\ell(x^{\ell+1}) = 0$ (Proposition 2.1 a)) it is obtained

$$\mu_i^\ell \langle \hat{s}_i^\ell, x^0 - x^{\ell+1}\rangle = \sum_{j\in\mathcal{B}_i^\ell} \langle \lambda_i^j s_i^j, x^0 - x^{\ell+1}\rangle \leq \mu_i^\ell c_i(x^0) \leq \mu_i^\ell \gamma < 0\,.$$

If $\mu_i^\ell = 0$, then $\lambda_i^j = 0$ for all $j \in \mathcal{B}_i^\ell$ and thus $\sum_{j\in\mathcal{B}_i^\ell} \langle \lambda_i^j s_i^j, x^0 - x^{\ell+1}\rangle = \mu_i^\ell c_i(x^0) = 0$. Hence,

$$\begin{aligned} 0 > \bar{\mu}^\ell \gamma \;&\geq\; \sum_{i\in\mathcal{I}_c} \mu_i^\ell c_i(x^0) \geq \sum_{i\in\mathcal{I}_c} \sum_{j\in\mathcal{B}_i^\ell} \langle \lambda_i^j s_i^j, x^0 - x^{\ell+1}\rangle + \langle s_X^\ell, x^0 - x^{\ell+1}\rangle \\ &=\; \bar{\mu}^\ell \langle \bar{s}^\ell, x^0 - x^{\ell+1}\rangle + \langle s_X^\ell, x^0 - x^{\ell+1}\rangle = \bar{\mu}^\ell \langle p^\ell, x^0 - x^{\ell+1}\rangle\,, \end{aligned}$$

and it was demonstrated that $0 > \gamma \geq -\|p^\ell\|\|x^0 - x^{\ell+1}\|$ for all ℓ. As a result, p^ℓ is bounded away from zero (because X is bounded and $x^\ell \in X$ for all ℓ) and therefore $\{\bar{\mu}^\ell\}$ is a bounded sequence. This concludes the proof. ∎

This result shows that Algorithm 2.1 increases the penalization parameter only finitely many times. As a consequence, ρ stabilizes and the study can rely on the theory of [18] to establish convergence of Algorithm 2.1.

Theorem 2.1. *Consider problem* (2.3) *with convex and continuous functions* ϕ, c_i :

$\mathbb{R}^{n_x} \to \mathbb{R}$. *Suppose $X \neq \emptyset$ is a bounded polyhedron, the sequence of prox-parameter $\{\tau^\ell\}$ satisfies $\tau_{\max} \geq \tau^\ell \geq \tau_{\min} > 0$ for all ℓ, and that the Slater condition holds. Let $\epsilon_0 = \epsilon_1 = \epsilon_2 = 0$ in Algorithm 2.1. Then*

(a) There exists an index set $K \subset \{1, 2, \ldots\}$ such that $\lim_{\ell \in K} \hat{e}^\ell = 0$, $\lim_{\ell \in K} d^\ell = 0$ and the whole sequence $\{\hat{x}^\ell\}$ converges to a minimum $\hat{x}$ of (2.3)*, and as a consequence $\lim_{\ell \in K} \max_{i \in \mathcal{I}_c} c_i(\hat{x}^\ell) \leq 0$. Moreover, $\lim_{\ell \in K} x^{\ell+1} = \hat{x}$.*

(b) Furthermore, the three sequences $\{\hat{s}_\phi^\ell\}_K$, $\{\hat{s}_i^\ell\}_K$ and $\{\mu_i^\ell\}_K$ (defined in Proposition 2.1) have cluster points. Let s_ϕ, s_i and $\bar{\mu}_i$ be arbitrary cluster points of these sequences, respectively. Then, $s_\phi \in \partial\phi(\hat{x})$, $s_i \in \partial c_i(\hat{x})$ if $\bar{\mu}_i > 0$, and $\bar{\mu}_i$ ($i \in \mathcal{I}_c$) satisfy the KKT system (2.4) *with $\bar{x} = \hat{x}$.*

Proof: Proposition 2.2 ensures that the penalization parameter ρ stabilizes after finitely many steps, and therefore the solution $\hat{x}^{\ell+1}$ of subproblem (2.6) also solves the QP (2.15) for all large enough ℓ (see the comments right after Proposition 2.1). As a result, after finitely many iterations Algorithm 2.1 boils down to be the classical proximal bundle algorithm applied to the problem of minimizing the penalized function f_ρ over X:

$$\min_{x \in X} f_\rho(x)\,, \quad \text{with } f_\rho \text{ given in (2.8).}$$

It follows from the analysis of the proximal bundle method (see Theorem 6.11 and § 7.1.1 of [18]) that there exists an index set K such that $\lim_{\ell \in K} \hat{e}^\ell = 0$, $\lim_{\ell \in K} d^\ell = 0$ and $\lim_{\ell \to \infty} \hat{x}^\ell = \hat{x}$ (see [18, Theorem 6.2 iii)]) is a solution of the above penalized problem. Moreover, since $\{\tau^\ell\}$ is a bounded sequence, it follows from (2.10) and the above results that

$$\lim_{\ell \in K} x^{\ell+1} = \lim_{\ell \in K} \hat{x}^\ell - \lim_{\ell \in K} \tau^\ell d^\ell = \hat{x}\,.$$

It remains to show that the cluster point $\hat{x}$ also solves (2.3). To this end it is just necessary to prove that $\hat{x}$ is feasible for (2.3), i.e., $c_i(\hat{x}) \leq 0$ for all $i \in \mathcal{I}_c$. Notice that Step 5 of the algorithm keeps the linearization of the last descent steps in the bundles $\mathcal{B}_i^\ell$, $i \in \mathcal{I}_c$. Then, as $x^{\ell+1}$ is feasible for (2.6) and $s_i^{\hat{\ell}} \in \partial c_i(\hat{x}^\ell)$, $c_i(\hat{x}^\ell) + \langle s_i^{\hat{\ell}}, x^{\ell+1} - \hat{x}^\ell \rangle \leq 0$ for all ℓ and all $i \in \mathcal{I}_c$ (because $\hat{x}^\ell = \hat{x}^{\hat{\ell}}$ by definition). By the Cauchy-Schwartz inequality it is obtained

$$c_i(\hat{x}^\ell) \leq \|s_i^{\hat{\ell}}\| \|x^{\ell+1} - \hat{x}^\ell\| = \|s_i^{\hat{\ell}}\| \|\tau^\ell d^\ell\|, \quad \forall\, i \in \mathcal{I}_c\,,\ \ell = 1, 2, \ldots\,.$$

Since X is a bounded set, the subgradients $s_i^{\hat{\ell}}$ of the convex functions c_i are also bounded, [39, Proposition 6.2.2]. By taking the limit with $\ell \in K$ in the above relation (and remembering that c_i are continuous functions) it is concluded that $c_i(\hat{x}) \leq 0$ for all $i \in \mathcal{I}_c$. Hence, the solution $\hat{x}$ of the penalized problem is feasible for (2.3), proving that $\hat{x}$ is also a solution to (2.3). This concludes the proof of item (a).

Existence of clusters points of the sequences in item (b) is ensured by the boundedness of the subdifferentials of convex functions on bounded convex sets X and Proposition 2.2. Let s_ϕ be a cluster point of $\{\hat{s}_\phi^\ell\}_K$, and let $K' \subset K$ be the index set gathering the iterations satisfying $s_\phi = \lim_{\ell \in K'} \hat{s}_\phi^\ell$. As the subsequence $\{\hat{s}_i^\ell\}_{K'}$ is also bounded, it has a cluster point s_i and therefore there exists an index set $K'' \subset K'$ such that $s_i = \lim_{\ell \in K''} \hat{s}_i^\ell$. By continuing with this reasoning a cluster point $\bar{\mu}_i$ of the bounded subsequence $\{\mu_i^\ell\}_{K''}$ and an index set $L \subset K''$ such that $\bar{\mu}_i = \lim_{\ell \in L} \mu_i^\ell$ are obtained. In summary, the index set L is such that

$$s_\phi = \lim_{\ell \in L} \hat{s}_\phi^\ell, \quad s_i = \lim_{\ell \in L} \hat{s}_i^\ell \quad \text{and} \quad \bar{\mu}_i = \lim_{\ell \in L} \mu_i^\ell.$$

As $\lim_{\ell \in K} \hat{e}^\ell = 0$ and $L \subset K$, it is concluded from Proposition 2.1 d) that $\lim_{\ell \in L} \hat{e}_\phi^\ell = 0$ and $\lim_{\ell \in L} \hat{e}_i^\ell = 0$ for all $i \in \mathcal{I}_c$ with $\bar{\mu}_i > 0$. It also follows from Proposition 2.1 d) that $\hat{s}_\phi^\ell \in \partial_{\hat{e}_\phi^\ell} \phi(\hat{x}^\ell)$ and $\hat{s}_i^\ell \in \partial_{\hat{e}_i^\ell} c_i(\hat{x}^\ell)$. Hence, by passing to the limit as $\ell \xrightarrow{L} \infty$ in the latter inclusions and recalling [40, Proposition 4.1.1] it is gotten $s_\phi \in \partial \phi(\hat{x})$ and $s_i \in \partial c_i(\hat{x})$ if $\bar{\mu}_i > 0$. (Notice that if K is a finite index set (so is L), then the same conclusion trivially holds from Proposition 2.1 e) and stopping test of Algorithm 2.1 with $\epsilon_0 = \epsilon_1 = \epsilon_2 = 0$.) From item (a) above, the cluster point $\hat{x}$ is an optimal solution to (2.3), and thus $c_i(\hat{x}) \leq 0$, $i \in \mathcal{I}_c$. It follows from the KKT conditions in Proposition 2.1 item a) that

$$0 = \mu_i^\ell c_i(x^{\ell+1}) = \mu_i^\ell c_i(\hat{x}^\ell - \tau^\ell d^\ell)\,.$$

Since τ^ℓ is bounded and $\lim_{\ell \in L} d^\ell = 0$, the limit in the above identity can be taken to conclude (by continuity of c_i) that $\bar{\mu}_i = 0$ whenever $c_i(\hat{x}) < 0$. Hence $\bar{\mu}_i c_i(\hat{x}) = 0$ for all $i \in \mathcal{I}_c$. Equation (2.10) gives

$$d^\ell - \left(\hat{s}_\phi^\ell + \sum_{i \in \mathcal{I}_c} \mu_i^\ell \hat{s}_i^\ell \right) = s_X^\ell \in N_X(x^{\ell+1})\,.$$

Notice that $\{s_X^\ell\}_L$ is a convergent sequence (because all of its ingredients indexed by L are convergent sequences as well). Since $\lim_{\ell \in L} x^{\ell+1} = \hat{x}$ and the normal cone of a convex set X is outer semicontinuous, Proposition 6.6 in [58] ensures that $\lim_{\ell \in L} s_X^\ell \in N_X(\hat{x})$, i.e.,

$$\lim_{\ell \in L} s_X^\ell = \lim_{\ell \in L} \left[d^\ell - \left(\hat{s}_\phi^\ell + \sum_{i \in \mathcal{I}_c} \mu_i^\ell \hat{s}_i^\ell \right) \right] = -\left(s_\phi + \sum_{i \in \mathcal{I}_c} \bar{\mu}_i s_i \right) \in N_X(\hat{x})\,.$$

This concludes the proof. ■

It was shown that Algorithm 2.1 asymptotically finds an optimal solution, subgradients and multipliers satisfying the KKT system (2.4) of problem (2.3). Hence, Algorithm 2.1

(that is not difficult to implement) appears as an interesting tool to be employed by OA algorithms to solve the (nonsmooth convex) OA's subproblems (2.1) and (2.2). This approach is formalized in the following chapter.

Chapter 3

Regularized OA algorithms for MINLP with nonsmooth convex functions

In Chapter 2, the focus was on the OA nonlinear subproblems and on a bundle method algorithm capable to solve them and providing appropriate subgradients. In this chapter, the attention will be on OA's MILP subproblems. The goal is to regularize the MILP subproblems in order to accelerate the OA algorithm. As solving MILP problems is a difficult task, the fewer MILP subproblem are solved the better.

3.1 Description of the method

In the sequel the regularized OA algorithms dealing with (possibly nonsmooth) convex MINLP problems will be presented. If the problem's functions are differentiable and there is no regularization, then the given algorithms boil down to the classical one. To this end, we recall the problem of interest:

$$f_{\min} := \min_{(x,y)\in X\times Y} f_0(x,y) \ \text{ s.t. } \ f_i(x,y) \le 0\,,\ i \in \mathcal{I}_c := \{1,\dots,m_f\}\,, \tag{3.1}$$

where $f_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}$, $i = 0, 1, \dots, m_f$, are convex functions, $X \subset \mathbb{R}^{n_x}$ is a simple convex set (e.g. a polyhedron), and $Y \subset \mathbb{Z}^{n_y}$ is an integer set. Both sets X and Y are assumed to be nonempty and bounded (as a result, Y contains finite number of points). Given $y^k \in Y$, one of the OA's (nonsmooth) subproblem is (as we have already seen in Chapter 1):

$$\min_{x\in X} f_0(x,y^k) \ \text{ s.t. } \ f_i(x,y^k) \le 0\,,\ i \in \mathcal{I}_c := \{1,\dots,m_f\}\,. \tag{3.2}$$

Suppose that problem (3.2) is feasible. As consequence at least a solution x^k is known. Under the Slater condition, there exists Lagrange multiplies μ_i^k satisfying the KKT system

$$\left\{\begin{array}{l} 0 \in \partial_x f_0(x^k, y^k) + \sum_{i \in \mathcal{I}_c(x^k)} \mu_i^k \partial_x f_i(x^k, y^k) + N_X(x^k, y^k) \\ \mu_i^k f_i(x^k, y^k) = 0\,, i \in \mathcal{I}_c(x^k) \\ \mu_i^k \geq 0,\ i \in \mathcal{I}_c(x^k)\,, \end{array}\right. \tag{3.3}$$

where $\mathcal{I}_c(x^k)$ is the set of active constraints at point x^k and $N_X(x^k, y^k)$ is a normal cone of X at point x^k and y^k is a fixed vector. From now on, the notation $N(x^k)$ will be used for this cone. The first line in the above system can be written as

$$0 = s_0^{x^k} + \sum_{i \in \mathcal{I}_c(x^k)} \mu_i^k s_i^{x^k} + s_X, \tag{3.4}$$

where $s_0^{x^k} \in \partial_x f_0(x^k, y^k)$, $s_i^{x^k} \in \partial_x f_i(x^k, y^k)$ and $s_X \in N(x^k)$. The vector $s_0^{x^k}$ belongs to $\mathbb{R}^{n_x}$. Another arbitrary vector $s_0^{y^k} \in \partial_y f_0(x^k, y^k)$ is needed, in a manner that $(s_0^{x^k}, s_0^{y^k})$ will be a subgradient of f_0 at point (x^k, y^k), to create a cut to the MILP master problem. In other words, a vector $s_0^{y^k} \in \mathbb{R}^{n_y}$ is required such that

$$f_0(x, y) \geq f_0(x^k, y^k) + \langle (s_0^{x^k}, s_0^{y^k}), (x - x^k, y - y^k) \rangle \tag{3.5}$$

holds for all $(x, y) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$. Analogously to the constraints, vectors $s_i^{y^k} \in \mathbb{R}^{n_y}$ are required such that

$$f_i(x, y) \geq f_i(x^k, y^k) + \langle (s_i^{x^k}, s_i^{y^k}), (x - x^k, y - y^k) \rangle \tag{3.6}$$

holds for all $(x, y) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$ and $i \in \mathcal{I}_c(x^k)$. Observe that the only part of the subgradient that requires satisfying KKT is the part relative to x. Any vectors $s_0^{y^k}$ and $s_i^{y^k}$ can be chosen such that (3.5) and (3.6) holds.

For a given fixed y^k, Lemma 3.1 below shows that the subproblem

$$\left\{\begin{array}{ll} \min\limits_{x \in X} & f_0(x^k, y^k) + \langle (s_0^{x^k}, s_0^{y^k}), (x - x^k, 0) \rangle \\ \text{s.t.} & f_i(x^k, y^k) + \langle (s_i^{x^k}, s_i^{y^k}), (x - x^k, 0) \rangle \leq 0\,, \quad i \in \mathcal{I}_c(x^k)\,. \end{array}\right. \tag{3.7}$$

has the same solution set of subproblem (3.2).

Lemma 3.1. *Consider the problem given by* (3.7)*. Suppose that* (3.3) *holds. Then x^k solution of* (3.2) *is an optimal solution for* (3.7) *and, moreover, $f_0(x^k, y^k)$ is its optimal value.*

Proof: To prove this result it is sufficient to show that

$$\langle (s_0^{x^k}, s_0^{y^k}), (x - x^k, 0)\rangle \geq 0,$$

for all $x \in X$ such that

$$f_i(x^k, y^k) + \langle (s_i^{x^k}, s_i^{y^k}), (x - x^k, 0)\rangle \leq 0\,,\ i \in \mathcal{I}_c(x^k). \tag{3.8}$$

Consider $x \in X$ such that (3.8) holds. Remember that by definition $f_i(x^k, y^k) = 0$ for all $i \in \mathcal{I}_c(x^k)$ and consequently

$$\langle s_i^{x^k}, x - x^k\rangle \leq 0\,,\ i \in \mathcal{I}_c(x^k). \tag{3.9}$$

Furthermore, there exists $s_X \in N(x^k)$ such that (3.4) holds. As X is a convex set, $\langle s_X, x - x^k\rangle \leq 0$ for all $x \in X$. So, by (3.4)

$$\begin{aligned}
\langle (s_0^{x^k}, s_0^{y^k}), (x - x^k, 0)\rangle &= \langle s_0^{x^k}, x - x^k\rangle \\
&= \langle -\sum_{i\in\mathcal{I}_c(x^k)} \mu_i^k s_i^{x^k} - s_X, x - x^k\rangle \\
&= -\sum_{i\in\mathcal{I}_c(x^k)} \mu_i^k \langle s_i^{x^k}, x - x^k\rangle - \langle s_X, x - x^k\rangle \geq 0\,,
\end{aligned}$$

where the first part is positive because $\mu_i^k \geq 0$ and holds (3.9). ∎

With this result, and assumption that (3.7) is feasible for all $y \in Y$, all the pairs (x^k, y^k) can be gathered and written a mixed integer linear programming equivalently to problem (3.2):

$$\left\{\begin{array}{lll}
\displaystyle\min_{(r,x,y)\in\mathbb{R}\times X\times Y} & r & \\
\text{s.t.} & f_0(x^j, y^j) + \langle (s_0^{x^j}, s_0^{y^j}), (x - x^j, y - y^j)\rangle \leq r\,, & \forall j \in T \\
& f_i(x^j, y^j) + \langle (s_i^{x^j}, s_i^{y^j}), (x - x^j, y - y^j)\rangle \leq 0\,, & \forall j \in T,\ i \in \mathcal{I}_c(x^j)\,,
\end{array}\right. \tag{3.10}$$

where T is given by

$$T = \{j|\ (3.2) \text{ is feasible and } x^j \text{ is an optimal solution.}\}$$

The constraints in problem (3.10) are known as optimality cuts.

Theorem 3.1. *Assume that the nonlinear problem* (3.2) *satisfies the Slater's condition and* (3.3) *holds for all* $j \in T$*. Then problem* (3.10) *is equivalent to problem* (3.1)*, in the sense that both have the same solution* $(\bar{x}, \bar{y})$ *with* $\bar{r} = f_0(\bar{x}, \bar{y})$*.*

Proof: Let $(\bar{x}, \bar{y})$ be an optimal solution of problem (3.1). Then $f_0(\bar{x}, \bar{y}) \leq f_0(x^j, y^j)$ for all $j \in T$. In particular, $f_0(\bar{x}, \bar{y}) \leq \bar{r}$, the optimal value of (3.10). But $(\bar{x}, \bar{y}) = (x^{\bar{j}}, y^{\bar{j}})$

for some $\bar{j} \in T$. As a result, $\bar{r} \leq r^{\bar{j}}$. Lemma 3.1 ensures that $r^{\bar{j}} = f_0(x^{\bar{j}}, y^{\bar{j}})$, and as a consequence

$$f_0(\bar{x}, \bar{y}) \leq \bar{r} \leq f_0(x^{\bar{j}}, y^{\bar{j}}) = f_0(\bar{x}, \bar{y}).$$

This completes the proof. ■

The previous result only holds if all nonlinear subproblems (3.2) are feasible and all the finitely many point $y^j \in Y$ are collected. Still this is not always the case. Now suppose that given y^k, problem (3.2) is infeasible. In this case, there exists at least an index $i \in \mathcal{I}_c$ such that $f_i(x, y^k) > 0$ for some $x \in X$. Once subproblem (3.2) is infeasible for a fixed y^k, the aim is to minimize infeasibility. A way to do that is to solve the subproblem:

$$\min_{x \in X} \sum_{i \in \mathcal{I}_c} \max\{f_i(x, y^k), 0\}\ . \tag{3.11}$$

Subproblem (3.11) has at least an optimal solution x^k because X is compact and the involved functions f_i are continuous. Let $\beta_k = \{i \in \mathcal{I}_c | f_i(x, y^k) > 0 \quad \text{for some} \quad x \in X\}$. For all $i \in \beta_k$ the vector $s_i^{x^k} \in \partial_x f_i(x^k, y^k)$ exists and it is also a subgradient in $\partial_x \max\{f_i(x^k, y^k), 0\}$; and for all $i \notin \beta_k$, $0 \in \partial_x \max\{f_i(x^k, y^k), 0\}$. Note that functions $\max\{f_i(\cdot, y^k), 0\}$ are convex and their domains are the entire space $\mathbb{R}^{n_x}$. Then, Theorem 7.4 in [62] yields

$$\partial_x \sum_{i \in \mathcal{I}_c} \max\{f_i(x^k, y^k), 0\} = \sum_{i \in \mathcal{I}_c} \partial_x \max\{f_i(x^k, y^k), 0\} = \sum_{i \in \beta_k} \partial_x \max\{f_i(x^k, y^k), 0\}.$$

As a result, the optimality condition of subproblem (3.11) writes as

$$0 \in \partial_x \sum_{i \in \beta_k} \max\{f_i(x^k, y^k), 0\} + N(x^k),$$

or alternatively

$$0 = \sum_{i \in \beta_k} s_i^{x^k} + s_X. \tag{3.12}$$

Using the same procedure as before, the existence of vectors $s_i^{y^k}$ can be assured and it belongs to $\partial_y f_i(x^k, y^k)$, $i \in \beta_k$ such that $(s_i^{x^k}, s_i^{y^k})$ is a subgradient of $\max\{f_i, 0\}$ at point (x^k, y^k), where x^k is a solution of (3.11).

Theorem 3.2. *The variable y^k which makes the NLP subproblem* (3.2) *infeasible does not satisfy the following constraints*

$$f_i(x^k, y^k) + \langle (s_i^{x^k}, s_i^{y^k}), (x - x^k, y - y^k) \rangle \leq 0,\ \forall i \in \beta_k \tag{3.13}$$

where $s_i^{x^k}$ satisfies (3.12), *$x \in X$ and $y \in Y$.*

Proof: Given y^k the solution x^k of problem (3.11) yields

$$\sum_{i\in\beta_k} \max\{f_i(x^k, y^k), 0\} > 0. \tag{3.14}$$

Suppose that the integer variable y^k is feasible to constraints (3.13). So there exists $\bar{x} \in X$ such that

$$f_i(x^k, y^k) + \langle (s_i^{x^k}, s_i^{y^k}), (\bar{x} - x^k, 0)\rangle \leq 0, \ \forall i \in \beta_k.$$

Simplifying,

$$f_i(x^k, y^k) + \langle s_i^{x^k}, \bar{x} - x^k\rangle \leq 0, \ \forall i \in \beta_k.$$

Summing for all $i \in \beta_k$ the inequality is written as

$$\sum_{i\in\beta_k} f_i(x^k, y^k) + \sum_{i\in\beta_k} \langle s_i^{x^k}, \bar{x} - x^k\rangle \leq 0.$$

Using (3.12)

$$\sum_{i\in\beta_k} \langle s_i^{x^k}, \bar{x} - x^k\rangle = \langle \sum_{i\in\beta_k} s_i^{x^k}, \bar{x} - x^k\rangle = \langle -s_X, \bar{x} - x^k\rangle \geq 0. \tag{3.15}$$

It becomes

$$\sum_{i\in\beta_k} f_i(x^k, y^k) \leq 0,$$

which leads to

$$\sum_{i\in\beta_k} \max\{f_i(x^k, y^k), 0\} = 0,$$

a contradiction with (3.14). ■

Consider the following index set

$$S = \{j|\ (3.2) \text{ is infeasible and } x^j \text{ is an optimal solution of } (3.11).\}$$

The next result shows that variables y^j, with $j \in S$, are eliminated from the feasible set of problem (3.1) by the feasibility cuts defined in (3.16) below.

Theorem 3.3. *Let j be an arbitrary index in S, $f_{\max}(x, y) = \sum_{i\in\beta_j} \max\{f_i(x, y), 0\}$, $s_{\max}^{x^j} = \sum_{i\in\beta_j} s_i^{x^j}$ and $s_{max}^{y^j} = \sum_{i\in\beta_j} s_i^{y^j}$, where $s_i^{x^j} \in \partial_x f_i(x^j, y^j)$ and $s_i^{y^j} \in \partial_y f_i(x^j, y^j)$. The cut*

$$f_{\max}(x^j, y^j) + \langle (s_{\max}^{x^j}, b_{\max}^{y^j}), (x - x^j, y - y^j)\rangle \leq 0 \tag{3.16}$$

excludes the variable $y^j \in Y$.

Proof: By putting $y = y^j$ in (3.16) the following inequality holds

$$\begin{aligned} 0 &\geq f_{\max}(x^j, y^j) + \langle s_{\max}^{x^j}, x - x^j \rangle \\ &= \sum_{i \in \beta_j} \max\{f_i(x^j, y^j), 0\} + \sum_{i \in \beta_j} \langle s_i^{x^j}, x - x^j \rangle \\ &> \sum_{i \in \beta_j} \langle s_i^{x^j}, x - x^j \rangle, \end{aligned}$$

which leads to a contradiction with (3.15). ∎

Note that to solve problem (3.11) we can start with cuts provided by (3.2). All the two situations that can occur when we fix an integer variable y were analyzed. By gathering all the possible optimality cuts in index set T and all the possible feasibility cuts in index set S, the following MILP problem can be written as

$$\left\{ \begin{array}{lll} \min\limits_{(r,x,y) \in \mathbb{R} \times X \times Y} & r & \\ \text{s.t.} & f_0(x^j, y^j) + \langle (s_0^{x^j}, s_0^{y^j}), (x - x^j, y - y^j) \rangle \leq r\,, & \forall j \in T \\ & f_i(x^j, y^j) + \langle (s_i^{x^j}, s_i^{y^j}), (x - x^j, y - y^j) \rangle \leq 0\,, & \forall j \in T,\ i \in \mathcal{I}_c(x^j) \\ & f_{\max}(x^l, y^l) + \langle (s_{\max}^{x^l}, s_{\max}^{y^l}), (x - x^l, y - y^l) \rangle \leq 0\,, & \forall l \in S\,, \end{array} \right. \tag{3.17}$$

which is equivalent (in terms of the optimal solutions and optimal value) to the MINLP (3.1).

In practice the index sets T and S are unknown. Iteratively subsets $T^k \subset T$ and $S^k \subset S$ are constructed gathering optimality and feasibility cuts, obtained up to iteration k:

$$T^k = \{j \leq k |\ (3.2) \text{ is feasible and } x^j \text{ is an optimal solution.}\}$$

$$S^k = \{j \leq k |\ (3.2) \text{ is infeasible and } x^j \text{ is an optimal solution of (3.11).}\}$$

Note that (x^j, y^j) is feasible to problem (3.1) for all $j \in T^k$. As a result, $f_{up}^k = \min\limits_{j \in T^k} f_0(x^j, y^j)$ is a upper bound for the optimal value of problem (3.1). As the involved function are convex, the following master problem provides a lower bound for (3.1).

$$\left\{ \begin{array}{lll} \min\limits_{(r,x,y) \in \mathbb{R} \times X \times Y} & r & \\ \text{s.t.} & r \leq f_{up}^k & \\ & f_0(x^j, y^j) + \langle (s_0^{x^j}, s_0^{y^j}), (x - x^j, y - y^j) \rangle \leq r\,, & \forall j \in T^k \\ & f_i(x^j, y^j) + \langle (s_i^{x^j}, s_i^{y^j}), (x - x^j, y - y^j) \rangle \leq 0\,, & \forall j \in T^k,\ i \in \mathcal{I}_c(x^j) \\ & f_{\max}(x^l, y^l) + \langle (s_{\max}^{x^l}, s_{\max}^{y^l}), (x - x^l, y - y^l) \rangle \leq 0\,, & \forall l \in S^k\,. \end{array} \right. \tag{3.18}$$

Note that the point (x^j, y^j) yielding f_{up}^k is feasible to above problem. In order to eliminate such a point (and as well as all previous iterates) the constraint $r \leq f_{up}^k$ is replaced by $r \leq f_{up}^k - \epsilon$, where $\epsilon > 0$ is an arbitrary small parameter. The size of problem (3.18) depends on the number of performed iterations, and can substantially grow making (3.18) a very difficult optimization problem.

In order to overcome this situation, the goal is to regularize the MILP subproblem in the sense that the minimum number of MILP as possible are solved allowing to stabilize the OA iterative process by computing cuts and determining new iterates nearby a region of the best known candidate solution (at iteration k) for (3.18). If visiting uninteresting regions are avoided, the OA algorithm may approximate better (and faster) the functions on regions containing global solutions. This may end up in solving less MILP subproblems and less nonlinear subproblems. To this end, it is proposed to add a norm $||\cdot||_\diamond$ to the objective function of (3.18):

$$\begin{cases} \min\limits_{(r,x,y)\in\mathbb{R}\times X\times Y} & r+\mu_k||(x,y)-(\hat{x}^k,\hat{y}^k)||_\diamond \\ \text{s.t.} & r\le f_{up}^k-\epsilon \\ & f_0(x^j,y^j)+\langle(s_0^{x^j},s_0^{y^j}),(x-x^j,y-y^j)\rangle\le r\,, & \forall j\in T^k \\ & f_i(x^j,y^j)+\langle(s_i^{x^j},s_i^{y^j}),(x-x^j,y-y^j)\rangle\le 0\,, & \forall j\in T^k,\ i\in\mathcal{I}_c(x^j) \\ & f_{\max}(x^l,y^l)+\langle(s_{\max}^{x^l},b_{\max}^{y^l}),(x-x^l,y-y^l)\rangle\le 0\,, & \forall l\in S^k\,, \end{cases} \tag{3.19}$$

where $\mu_k>0$ is a parameter controlling the influence of the norm. In this formulation the pair $(\hat{x}^k,\hat{y}^k)$, known as stability center [16], can be the current iterate (x^k,y^k) or the pair yielding f_{up}^k. The norm and parameter μ_k can be chosen freely. For instance, $||\cdot||_\diamond=||\cdot||_1$ or $||\cdot||_\diamond=||\cdot||_\infty$ leads (3.19) to a MILP. On the other hand, the choice $||\cdot||_\diamond=||\cdot||_2$ results on a MIQP, which is in general more difficult to solve than a MILP. The choice of this norm depends on the structure of the problem. The usefulness of using regularization in MINLP has been evidenced in [16, 68, 78]. However, regularized techniques have not been employed so far in OA algorithms (except in [20], based in this work). The following regularized OA algorithm is proposed.

Algorithm 3.1. A REGULARIZED OUTER APPROXIMATION ALGORITHM

Step 0. *(Initialization) Choose* $y^0\in Y$, $\epsilon>0$, *a norm* $||\cdot||_\diamond$ *and set* $f_{up}^{-1}=\infty$,$T^{-1}=S^{-1}=\emptyset$, $k=0$.

Step 1. *(NLP) Apply Algorithm 2.1 to subproblem* (3.2). *If along the iterative process the corresponding QP* (2.6) *is infeasible, go to Step 2. Otherwise, let* $x^k\in X$, *and* $s_i^{x^k}\in\partial_x f_i(x^k,y^k), i=0,1,\ldots,m_f$ *be returned by Algorithm 2.1. Compute arbitrary vectors* $s_i^{y^k}\in\partial_y f_i(x^k,y^k), i=0,1,\ldots,m_f$, *set* $T^k=T^{k-1}\cup\{k\}$ *and* $S^k=S^{k-1}$. *Update the upper bound* $f_{up}^k=\min\{f_{up}^{k-1},f_0(x^k,y^k)\}$ *and go to Step 3.*

Step 2. *(NLP feasibility) Solve the feasibility subproblem* (3.11) *with Algorithm 2.1 to obtain* $x^k\in X, s_{\max}^{x^k}\in\partial_x f_{\max}(x^k,y^k)$ *with* $f_{\max}(\cdot,y^k):=\max\limits_{i\in\mathcal{I}_c}\{f_i(\cdot,y^k),0\}$. *Compute an arbitrary vector* $s_{\max}^{y^k}\in\partial_y f_{\max}(x^k,y^k)$, *set* $S^k=S^{k-1}\cup\{k\}$ *and* $T^k=T^{k-1}$.

Step 3. *(Integer trial point) If* (3.19) *is infeasible, go to Step 4. Otherwise, let y^{k+1} be the $y-$part of solution of problem* (3.19)*. Set $k = k + 1$ and go back to Step 1.*

Step 4. *(Termination) If $T^k = \emptyset$, then MINLP* (3.19) *is infeasible. Otherwise, return the ϵ-solution (x^*, y^*), with (x^*, y^*) the pair of points yielding $f_{\text{up}}^k = f_0(x^*, y^*)$. Terminate the algorithm.*

Since the feasible set of the bundle method QP problem (2.6) is an outer approximation of the feasible set of (2.3), if the former is empty so is the latter. This is why the above algorithm moves to the feasibility problem whenever the bundle method QP is empty. Observe that in this case the nonlinear subproblem is solved using Algorithm 2.1 because its provides subgradients that satisfy KKT conditions as explain before. Moreover, the master problem yielding y^{k+1} is the regularized one (3.19). In the step 0 of Algorithm 3.1, we can choose a parameter μ_0 and a rule to updated this parameter. In this work, we set this parameter as a constant, but a iterative process can be used as well.

3.2 Convergence analysis

The convergence analysis of Algorithm 3.1 is based on [26]. As the number of integer variables of problem (3.1) is finite, it is enough to show that OA Algorithm 3.1 does not repeat points. As a result, the algorithm finds an optimal solution of problem (3.1) (if any) in finitely many steps, or proves that the problem is infeasible.

Lemma 3.2. *Let C^*_{MINLP} be the solution set of problem* (3.1) *and C_{MILP} be the feasible set of problem* (3.19)*. Given $\epsilon > 0$ in Algorithm 3.1, let $\bar{f}_0$ be the optimal value of problem* (3.1) *and f^k_{up} as in the algorithm. If $f^k_{up} - \epsilon \geq \bar{f}_0$ then $C^*_{MINLP} \subset C_{MILP}$.*

Proof: If $C^*_{MINLP} = \emptyset$ then the result trivially follows. Assuming that $\emptyset \neq C^*_{MINLP} \ni (\bar{x}, \bar{y})$. As $f_i, i = 0, \ldots, m_f$ are convex functions, the following inequalities holds:

$$
\begin{aligned}
f^k_{up} - \epsilon \;\; &\geq \;\; \bar{f}_0 = f_0(\bar{x}, \bar{y}) \geq f_0(x^j, y^j) + \langle (s_0^{x^j}, s_0^{y^j}), (\bar{x} - x^j, \bar{y} - y^j) \rangle, \quad j \in T^k \\
0 \;\; &\geq \;\; f_i(\bar{x}, \bar{y}) \geq f_i(x^j, y^j) + \langle (s_i^{x^j}, s_i^{x^j}), (\bar{x} - x^j, \bar{y} - y^j) \rangle \,, \quad \forall j \in T^k, \, i \in \mathcal{I}_c(x^j) \\
0 \;\; &\geq \;\; f_{\max}(\bar{x}, \bar{y}) \geq f_{\max}(x^l, y^l) + \langle (s_{\max}^{x^l}, s_{\max}^{y^l}), (\bar{x} - x^l, \bar{y} - y^l) \rangle \,, \quad \forall l \in S^k \,.
\end{aligned}
$$

It was then demonstrated that $(\bar{x}, \bar{y}) \in C_{MILP}$. As $(\bar{x}, \bar{y})$ is an arbitrary point in C^*_{MINLP}, the proof is complete. ■

Theorem 3.4. *Suppose that subproblem* (3.2) *satisfies the Slater's condition, the chosen subgradients at Step* 2 *of Algorithm 3.1 satisfy the KKT conditions and $|Y| < \infty$. Then the algorithm terminates after finitely many steps either with an ϵ-solution of* (3.1) *or proving that* (3.1) *is infeasible.*

Proof: In order to prove that Algorithm 3.1 converges in a finite number of steps it is sufficient to show that any $y \in Y$ provides by the algorithm does not repeat, because the set Y has only finitely many points. At iteration k, let $(\bar{x}, \bar{y}, \bar{r})$ be a solution of problem (3.19). Suppose $\bar{y} = y^j$ for some $j \in S^k$ or $j \in T^k$. By Theorem 3.3, if $j \in S^k$ then the inequality

$$f_{\max}(x^j, y^j) + \langle (s_{\max}^{x^j}, s_{\max}^{y^j}), (x - x^j, y - y^j) \rangle \leq 0,$$

excludes the variable $y^j \in Y$. That means $j \in T^k$. As $(\bar{x}, \bar{y}, \bar{r})$ is a solution of problem (3.19) and $\bar{y} = y^j$, then $\bar{y}$ in (3.19) can be replaced and the following inequalities hold

$$\begin{aligned} &\bar{r} \leq f_{up}^k - \epsilon \leq f_0(x^j, y^j) - \epsilon \\ &f_0(x^j, y^j) + \langle (s_0^{x^j}, s_0^{y^j}), (\bar{x} - x^j, 0) \rangle \leq \bar{r} \\ &f_i(x^j, y^j) + \langle (s_i^{x^j}, s_i^{x^j}), (\bar{x} - x^j, 0) \rangle \leq 0 \end{aligned} \tag{3.20}$$

As subgradients satisfy KKT conditions the inner product $\langle s_0^{x^j}, \bar{x} - x^j \rangle \geq 0$ holds (see proof of Lemma 3.1). So by the second inequality of (3.20) $f_0(x^j, y^j) \leq \bar{r}$. By the first inequality of (3.20), $f_0(x^j, y^j) \leq f_0(x^j, y^j) - \epsilon$ which is a contradiction because $\epsilon > 0$. Therefore $j \notin T^k \cup S^k$ and the previous integer variable does not repeat in the OA algorithm. As $|Y| < \infty$, the algorithm will stop after finitely many steps.

Now it will be proved that the algorithm either provides a ϵ-solution or proves that (3.1) is infeasible. Suppose that Algorithm 3.1 stops at iteration k. If $T^k = \emptyset$ then problem (3.1) is infeasible. Now assume $T^k \neq \emptyset$, i.e., problem (3.1) is feasible. When the algorithm stops at Step 3 with subproblem (3.19) infeasible, it follows from Lemma 3.2 and the fact that $C^*_{MINLP} \neq \emptyset$ that $f_{up}^k - \bar{f}_0 < \epsilon$, i.e., the point $(x^{\bar{j}}, y^{\bar{j}})$ yielding f_{up}^k is a $\epsilon-$solution. This concludes the proof. ∎

Note that the norm does not interfere in the convergence of OA algorithm. The next chapter addresses nonsmooth convex MINLP problems with chance constraints. The numerical results about this theory will be present in Chapter 5.

Chapter 4

Nonsmooth convex MINLP with chance constraints

In this chapter we considered MINLPs with chance constraints. In Section 4.1 a few well-known results on Chance-Constrained Programming are revisited. Section 4.2 is dedicated to Copulae, which are multivariate functions that approximate probability functions. Chance-Constrained MINLP (CCMINLP) problems are considered in Section 4.3.

4.1 Chance constraints

In this thesis, stochastic optimization problems where the randomness appears only in the constraints are studied. For instance, constrains represented by

$$h_i(x, \xi) \geq 0, \forall i = 1, \ldots, s$$

where $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^s$ is a mapping having generalized concavity properties on a given level set and $\xi \in \mathbb{R}^m$ is a random vector. One strategy that is widely employed to deal with problems of this class is chance-constrained programming, which replaces the above stochastic constraint by the probability one

$$P[h(x, \xi) \geq 0] \geq p, \tag{4.1}$$

where P is the probability measure associated to ξ, and $p \in (0, 1]$ is a given parameter. Chance constraints appear in several real life problems such as water management, telecommunications, electricity network expansion, mineral blending, chemical engineering and others [36, 50, 55, 66, 71].

In a general term, a stochastic optimization problem involving chance constraints is written as

$$\min_{x\in X} \phi(x) \;\text{ s.t. }\; P[h(x,\xi) \geq 0] \geq p\,. \tag{4.2}$$

Function $\phi : \mathbb{R}^n \to \mathbb{R}$ is assumed to be convex but not necessarily differentiable and $X \neq \emptyset$ is a given convex set, that does not depend on uncertain parameters. Basically, a point x is feasible for problem (4.2) if the system of equations $h(x, \xi) \geq 0$ is satisfied with probability at least p. It is important to mention that the function $P[h(x, \xi) \geq 0]$ may fail to be differentiable even when h is smooth. This is the case when ξ follows a multivariate normal distribution having a singular covariance matrix [70].

Chance constraints problems have been introduced by Chernes, Cooper and Symonds in 1958 in the papers [13] and [14]. The first proposed method was based on individual chance constraints

$$\min_{x\in X} \phi(x) \;\text{ s.t. }\; P[h_i(x,\xi) \geq 0] \geq p, \forall i = 1, \ldots, s\,. \tag{4.3}$$

Observe that problem (4.2) and (4.3) are different from each other. The constraint in (4.2) is called joint chance constraint and the ones in (4.3) are individual chance constraints. Miller and Wagner [49] investigated problem (4.3) where the stochastic components are independent. The general case, where the random vector could have dependent components was introduced by Prékopa in the papers [53, 54].

Probability constraints lead to some difficulties: the first one is that evaluating the probability function involve, in general, computing numerically a multidimensional integral. Depending on the dimension of the random vector ξ, the task of evaluating the probability constraint becomes computationally very expensive. There are at least two known manners to overcome this difficulty:

- considering inexact values for the probability function, an approach already investigated in [68];

- approximating the probability by a simpler function, also studied by [61, 66, 69, 71].

In this work the second strategy is employed and the probability function is approximated by a suitable copula, as discussed in Section 4.3 below.

Another difficulty in dealing with chance-constrained programming is that the probability constraint can yield a nonconvex feasible set even if function h in (4.3) is concave. The references [37, 38, 55, 72] have addressed this issue for a broad class of probability measures. In what follows some main results on convexity of the set issued by probability functions are reviewed.

4.1.1 Generalized convexity of chance constraint

In this section convexity of the following set

$$M(p) = \{x \in \mathbb{R}^n | P[h(x,\xi) \geq 0] \geq p\} \tag{4.4}$$

is reviewed. To this end, it is necessary to have some results on generalized concavity and its properties . The following useful definition is classic.

Definition 4.1. *Let $\alpha \in [-\infty, \infty]$ and $m_\alpha : \mathbb{R}_+ \times \mathbb{R}_+ \times [0,1] \to \mathbb{R}$ be defined as*

$$m_\alpha(a,b,\lambda) = 0 \text{ if } ab = 0,$$

and for $a > 0, b > 0, \lambda \in [0,1]$:

$$m_\alpha(a,b,\lambda) = \begin{cases} a^\lambda b^{1-\lambda} \text{ if } \alpha = 0 \\ \max(a,b) \text{ if } \alpha = \infty \\ \min(a,b) \text{ if } \alpha = -\infty \\ (\lambda a^\alpha + (1-\lambda) b^\alpha)^{\frac{1}{\alpha}} \text{ otherwise}\cdot \end{cases}$$

The function defined above is used to generalize concavity. The following lemma is given in [62].

Lemma 4.1. *The function $\alpha \mapsto m_\alpha(a,b,\lambda)$ is nondecreasing and continuous.*

The extension of concavity follows from the next definition.

Definition 4.2. *Consider a nonnegative function f defined on some convex set $\Omega \subset \mathbb{R}^n$. Then f is called $\alpha-$concave ($\alpha \in [-\infty, \infty]$) if only if*

$$f(\lambda x + (1-\lambda)y) \geq m_\alpha(f(x), f(y), \lambda) \; \forall x, y \in \Omega, \lambda \in [0,1],$$

where m_α is the function in Definition 4.1.

If $\alpha = 0$, then function f is called *log-concave* because $\log f(\cdot)$ is a concave function. If $\alpha = 1$, then f is concave; if $\alpha = -\infty$ then f is a quasi-concave function. If $\beta \leq \alpha$ and the function f is $\alpha-$concave, then by Lemma 4.1 f is $\beta-$concave. The same definition above can be used to define generalized concavity for probability measure.

Definition 4.3. *Consider P a probability measure defined on some measurable convex set $\Omega \subset \mathbb{R}^n$. Then P is called $\alpha-$concave ($\alpha \in [-\infty, \infty]$) if only if*

$$P[\lambda A + (1-\lambda)B] \geq m_\alpha(P[A], P[B], \lambda),$$

for all measurable sets A and B subsets of Ω and $\lambda \in [0,1]$. Here $\lambda A + (1-\lambda)B$ is the Minkowski sum, that is, $\lambda A + (1-\lambda)B = \{\lambda x + (1-\lambda)y, x \in A, y \in B\}$.

Given a random vector $\xi \in \mathbb{R}^m$, it has $\alpha-$concave distribution if the margins P_{ξ_i} induced by ξ_i is $\alpha-$concave. It is important to know the link between the margins and the cumulative distribution function. This is given by the following lemma.

Lemma 4.2. *If a random vector ξ induces a $\alpha-$concave probability measure in $\mathbb{R}^m$, then its cumulative distribution function F_ξ is a $\alpha-$concave function.*

Proof: See Lemma 4.12 in [62]. ■

An important result says about the margins of ξ with the distribution of ξ.

Lemma 4.3. *If a random vector ξ has independent components with log-concave marginal distribution, then ξ has a log-concave distribution.*

Proof: See Lemma 4.13 in [62]. ■

Note that the function in (4.1) is a composite map. Therefore to know about generalized concavity of composite functions the following theorem is necessary.

Theorem 4.1. *If f is a concave function defined on a convex set $\Omega \subset \mathbb{R}^n$ and $g : \mathbb{R} \to \mathbb{R}$ is a nonnegative nondecreasing $\alpha-$concave function, $\alpha \in [-\infty, \infty]$, then the function $g \circ f$ is $\alpha-$concave.*

Proof: See Theorem 4.20 in [62]. ■

When functions $h_j, j = 1, \ldots, s$ are considered as being quasi-concave, the next theorem assure that function (4.5) is $\alpha-$concave on the set (4.6) .

Theorem 4.2. *Let functions $h_j : \mathbb{R}^n \times \mathbb{R}^m, j = 1, \ldots, s$ be quasi-concave. If $\xi \in \mathbb{R}^m$ is a random vector that has $\alpha-$concave probability distribution, then the function*

$$G(x) = P[h_j(x,\xi) \geq 0, j = 1, \ldots, s] \tag{4.5}$$

is $\alpha-$concave on the set

$$D = \{x \in \mathbb{R}^n : \exists \xi \in \mathbb{R}^m \text{ such that } h_j(x,\xi) \geq 0, j = 1, \ldots, s\}. \tag{4.6}$$

Proof: See Theorem 4.39 in [62]. ■

Note that function (4.5) is $\alpha-$concave at variable x, and then if function is $\alpha-$concave in the other coordinate, the convexity of set (4.4) can be ensured. This result is given by the next corollary.

Corollary 4.1. *Assuming that functions,* $h_j(\cdot,\cdot), j = 1,\ldots,s$ *are quasi-concave jointly in both arguments and that* $\xi \in \mathbb{R}^m$ *is a random vector that has* $\alpha-$*concave probability distribution then* (4.4) *is a convex and closed set.*

Results about convexity are important because many random vector ξ have a probability distribution which is $0-$ concave. Consequently in order to use outer-approximation algorithms in problems with chance constraints, the "log" function must be applied in the probability function and then the composite function (multiplied by -1) is convex. The function h can be separable and the random vector ξ has independent components. In this case, the joint probability constraint can be written as a product of individual chance constraint and the problem become easier to solve. In some applications, the function h is not separable and even if it is, the random variable ξ has dependent components in most of the time. Even when the set $M(p)$ is convex, approximating it by linearizations of the probability function is a difficult task due to the following reasons:

- evaluating the probability function $P[h(x,\xi) \geq 0]$ involves computing numerically a multidimensional integral;
- computing a subgradient of $P[h(x,\xi) \geq 0]$ requires evaluating (numerically) m integrals of dimension $m-1$, [70].

When the dimension of the random vector ξ is large, computing a linearization for the probability constraint is too time consuming. To overcome this difficulty, the probability P can be approximated by an appropriate copula $\mathbb{C}$, modeling the dependence of the components of ξ. Some results about copulae are reviewed in the next section.

4.2 Copulae: a bird's eye view

In this section separable chance constraints are considered, that is, function h in (4.1) is given by $h(x,\xi) = g(x) - \xi$, where $g : \mathbb{R}^n \to \mathbb{R}^m$. When dealing with chance-constrained programs it is, very often, impossible to get an explicit formula for the probability measure P because the jointly distribution of ξ variable is unknown. In what follows, the random variable $\xi \in \mathbb{R}^m$ will supposed to have known marginal distributions $F_{\xi_1},\ldots,F_{\xi_m}$. This is a weaker assumption than assuming that the joint distribution of ξ is known. In order to model the dependence among theses marginals a copula function will be employed.

The concept of copula was introduced by Sklar [63] in 1959, when he was studying the relationship between a multidimensional probability function and its lower dimensional marginals.

Definition 4.4. *An $m-$dimensional copula is a function $\mathbb{C} : [0, 1]^m \to [0, 1]$ that satisfies the following properties:*

i) $\mathbb{C}(1, \ldots, 1, u, 1, \ldots, 1) = u \ \forall u \in [0, 1]$.

ii) $\mathbb{C}(u_1, \ldots, u_{i-1}, 0, u_{i+1}, \ldots, u_m) = 0$.

iii) $\mathbb{C}$ *is quasi monotone on* $[0, 1]^m$.

In other words, the above definition means that $\mathbb{C}$ is a $m-$dimensional distribution function with all univariate marginals being uniform in the interval $[0, 1]$. The item (iii) means that the $\mathbb{C}-$volume of any box in $[0, 1]^m$ is nonnegative (see [52] for more details).

Given a random vector ξ with known margins $F_{\xi_i}, i = 1, \ldots, m$, an important tool proved by Sklar [63] is a theorem that assures the existence of a copula that approximates the cumulative distribution F. This theorem only assures the existence of a copula and is reported below.

Theorem 4.3. *Let F_ξ be a $m-$dimensional distribution function with marginals $F_{\xi_1}, F_{\xi_2}, \ldots, F_{\xi_m}$. Then there exists a $m-$dimensional copula $\mathbb{C}$ such for all $z \in \mathbb{R}^m$,*

$$F_\xi(z_1, z_2, \ldots, z_m) = \mathbb{C}(F_{\xi_1}(z_1), F_{\xi_2}(z_2), \ldots, F_{\xi_m}(z_m)). \tag{4.7}$$

If $F_{\xi_i}, i = 1, \ldots, m$ are continuous, then $\mathbb{C}$ is unique. Otherwise, $\mathbb{C}$ is uniquely determined in the image of F_ξ. Conversely, if $\mathbb{C}$ is a copula and $F_{\xi_i}, \ldots, F_m$ are distribution functions, then the function F_ξ defined by (4.7) is a $m-$dimensional distribution function with marginals $F_{\xi_1}, \ldots, F_{\xi_m}$.

In the above theorem, functions $F_{\xi_i}, i = 1, \ldots, m$ can be different. Observe that this theorem is not constructive, it just ensures the existence of a copula associated to the distribution $F_\xi(z)$. In most of the cases, a copula providing the equality

$$\mathbb{C}(F_{\xi_1}(z_1), \ldots, F_{\xi_m}(z_m)) = F_\xi(z)$$

is unknown. One exception is when the random vector is independent, whose associated copula is the product copula:

$$\mathbb{C}(u_1, \ldots, u_m) = u_1 u_2 \cdots u_m.$$

The problem of choosing/estimating a suitable copula has been receiving (from the statistical community) much attention in the last few years, [15, 52]. As shown in books [25, 52], there are many copulae in the literature.

Any copula $\mathbb{C}$ can be bounded by the functions $W^m(u_1, u_2, \ldots, u_m) = \max\{u_1 + u_2, \ldots u_m - m + 1, 0\}$ and $M^m(u_1, u_2, \ldots, u_m) = \min\{u_1, u_2, \ldots, u_m\}$. Theses bounds are expressed by the following theorem, whose proof can be found in [52].

Theorem 4.4. *If $\mathbb{C}$ is a copula, then for all vector $u = (u_1, \ldots, u_m)$ belonging to the domain of $\mathbb{C}$ the following inequality holds*

$$W^m(u) \leq \mathbb{C}(u) \leq M^m(u).$$

Functions W^m and M^m are known as *Frechet-Hoeffiding* bounds. The map M^m is a copula for any dimension, and W^m is a copula for dimension $m = 2$ only.

4.3 Chance-constrained MINLP problems

In recent years, the stochastic programming community have been witnessed a great development in optimization methods for dealing with stochastic programs with mixed-integer variables [8]. However, there are only few works on chance-constrained programming with mixed-integer variables, [3, 16, 64, 74].

In this section, the problem of interest consists in nonsmooth convex mixed-integer nonlinear programs with chance constraints (CCMINLP). These class of problems can be solved by employing the outer-approximation technique presented in Chapter 3. In general, OA algorithms require solving less MILP subproblems than extended cutting-plane algorithms [76], therefore the former class of methods is preferable than the latter one. This justifies why we have chosen the former class of methods to deal with problems of the type

$$\begin{array}{lll} \min\limits_{(x,y)\in X\times Y} & f_0(x,y) & \\ \text{s.t.} & f_i(x,y) \leq 0, \quad i = 1, \ldots, m_f - 1 \\ & P[h((x,y),\xi) \geq 0] \geq p, & \end{array} \tag{4.8}$$

where

- $f_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}$, $i = 0, \ldots, m_f - 1$, are convex but possibly nonsmooth functions;
- $X \subset \mathbb{R}^{n_x}$ is a polyhedron;
- $Y \subset \mathbb{Z}^{n_y}$ contains only integer variables;
- both X and Y are compacts sets;
- $h((x,y),\xi) = g(x,y) - \xi$, where function $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^m$;
- $\xi \in \mathbb{R}^m$ is the random vector;

- $p \in (0,1)$ is a given parameter;
- P is the probability measure associated to the random vector ξ.

Furthermore, g is assumed to be a concave function and P a $0-$concave distribution (thus P is $\alpha-$concave for all $\alpha \leq 0$). Some examples of distribution functions that satisfies the $0-$concavity property are the well-known multidimensional Normal, Log-normal, Gamma and Dirichlet distributions [55]. Under these assumptions, the following function is convex [Theorem 4.2]

$$f_{m_f}(x,y) = \log(p) - \log(P[h((x,y),\xi) \geq 0)]) = \log(p) - \log(P[g(x,y) \geq \xi]). \tag{4.9}$$

As a result, (4.8) is a convex (but possibly nonsmooth) MINLP problem fitting notation (3.1) of Chapter 3:

$$f_{\min} := \min_{(x,y)\in X\times Y} f_0(x,y) \;\text{ s.t. }\; f_i(x,y) \leq 0\,,\; i \in \mathcal{I}_c := \{1,\ldots,m_f\}\,. \tag{4.10}$$

In addition to the difficulties present in MINLP models, the above problem has two more complications: the involved functions can be nondifferentiable and, mainly, f_{m_f} encompasses a probability function. Consequently, since problem (4.10) is a convex MINLP, Algorithm 3.1 developed in Chapter 3 can be applied.

Let us now consider the OA's nonlinear subproblem (3.2) with the last function representing a joint probability constraint (the analysis given below is analogous for the feasibility problem (3.11)). Given a fixed $y^k \in Y$, the nonlinear subproblem (3.2) with f_{m_f} replaced by (4.9) becomes

$$\begin{array}{ll} \min\limits_{x\in X} & f_0(x,y^k) \\ \text{s.t.} & f_i(x,y^k) \leq 0, \quad i = 1,\ldots,m_f - 1 \\ & \log(p) - \log(P[g(x,y^k) \geq \xi]) \leq 0. \end{array} \tag{4.11}$$

4.3.1 Chance-constrained MINLP problems: an approximation using Copulae

Due to the probability function $P[g(x,y) \geq \xi]$, evaluating the last constraint in problem (4.11) and computing its subgradient is a difficult task: as previously explained in Section 4.1, computing a subgradient of $P[g(x,y) \geq \xi]$ requires numerically solving m integrals of dimension $m-1$. If the dimension m of ξ is too large, then creating a cut for function $\log(p) - \log(P[g(x,y) \geq \xi])$ is computationally challenging. In this situation, it makes sense to replace the probability measure by a simpler function. In this manner, this work proposes to approximate the hard chance constraint $P[g(x,y) \geq \xi] \geq p$ by a copula $\mathbb{C}$:

$$\mathbb{C}(F_{\xi_1}(g_1(x,y)), F_{\xi_2}(g_2(x,y)), \ldots, F_{\xi_m}(g_m(x,y)) \geq p.$$

By applying "log" in the inequality above the following function is obtained

$$f_m(x,y) = \log(p) - \log \mathbb{C}(F_{\xi_1}(g_1(x,y)), F_{\xi_2}(g_2(x,y)), \ldots, F_{\xi_m}(g_m(x,y)), \tag{4.12}$$

where F_{ξ_i} is the marginal probability distribution of $F_\xi(z) = P[z \geq \xi]$, which is assumed to be known. The function given by (4.12) is well defined by Sklar's theorem [Theorem 4.3]. If $\mathbb{C}$ is $0-$concave, then (4.11) can be approximated by the convex MINLP

$$\begin{array}{ll} \min\limits_{x \in X} & f_0(x, y^k) \\ \text{s.t.} & f_i(x, y^k) \leq 0, \quad i = 1, \ldots, m_f - 1 \\ & \log(p) - \log(\mathbb{C}(F_{\xi_1}(g_1(x,y^k)), F_{\xi_2}(g_2(x,y^k)), \ldots, F_{\xi_m}(g_m(x,y^k))) \leq 0. \end{array} \tag{4.13}$$

In order to have a good approximation of chance constraint it is mandatory that

$$\mathbb{C}(F_{\xi_1}(g_1(x,y)), F_{\xi_2}(g_2(x,y)), \ldots, F_{\xi_m}(g_m(x,y)) \approx F_\xi(g_1(x,y), g_2(x,y), \ldots, g_m(x,y)),$$

for all (x,y) in a neighborhood of the solution set of problem (4.10). An appropriate copula must be chosen in a way that

- approximates well the underlying probability function;
- has generalized concavity properties so that after a simple transform (e.g. log) the resulting function is concave, and the CCMINLP problem becomes convex.

In the next section we present a family of copulas that satisfy the above requirements (the first condition is verified numerically whereas the second one is asserted by Theorem 4.2.)

4.3.2 Zhang's copulae

In order to ensure convexity of the underlying MINLP problem, suitable copulae must be chosen (e.g. concave and $\alpha-$ concave copulae with $\alpha \leq 0$). Consider any copula $\mathbb{C}$. By applying the "log" function in this copula, by Theorem 4.4 the following inequality is obtained

$$\log W^m(u) \leq \log \mathbb{C}(u) \leq \log M^m(u).$$

As the only concave copula is $M^n(u)$ (see [52, § 3.26]), in order that the copula $\mathbb{C}$ must be log concave this copula should be

$$\mathbb{C}(u) = K \cdot M^m(u)$$

for all natural K. In other words, the copula $\mathbb{C}$ is a product of copula M^n. Using the logarithm property $a \log b = \log b^a$, the number K can be decomposed, for instance, $K = K_1 \cdot K_2$ and the copula can be written as $\mathbb{C}(u) = K_1 \cdot M^m(u)^{K_2}$. This means that the copula with this property must be the product of powers copulae M^n.

There is a family of copula with this property, introduced by Zhang [79]. The family is given by

$$\mathbb{C}(u_1, \ldots, u_m) = \prod_{j=1}^{r} \min_{1 \le i \le m} (u_i^{a_{j,i}}), \tag{4.14}$$

where $a_{j,i} \geq 0$ and $\displaystyle\sum_{j=1}^{r} a_{j,i} = 1$ for all $i = 1, \ldots, m$. Different choices of parameters $a_{j,i}$ give different copulae, all of them nonsmooth functions, but with subgradient easily computed via chain rule. The next result shows that this family of copula is a log concave.

Proposition 4.1. *Let $\xi \in \mathbb{R}^m$ be a random vector with all marginals $F_{\xi_i}, i = 1, \ldots, m$ being $0-$concave functions. Suppose that $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^m$ is a concave function. Consider a Zhang's Copula $\mathbb{C}$ given in (4.14) for a certain choice of parameters $a_{j,i}$. Then*

$$\mathbb{C}(F_{\xi_1}(g_1(x, y)), F_{\xi_2}(g_2(x, y)), \ldots, F_{\xi_m}(g_m(x, y)))$$

is $\alpha-$concave for $\alpha \leq 0$.

Proof: Given a pair $(x, y) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$ we set $z = (x, y)$ to simplify the notation. Let $z_1 = (x_1, y_1), z_2 = (x_2, y_2)$ and $z = \lambda z_1 + (1 - \lambda)z_2$ with $\lambda \in [0, 1]$. As the function g is concave, then for all $i = 1, \ldots, m$

$$g_i(\lambda z_1 + (1 - \lambda)z_2) \geq \lambda g_i(z_1) + (1 - \lambda)g_i(z_2). \tag{4.15}$$

As F_{ξ_i}, $i = 1, \ldots, m$, are increasing functions, by applying F_{ξ_i} to inequality (4.15) it becomes

$$F_{\xi_i}(g_i(\lambda z_1 + (1 - \lambda)z_2)) \geq F_{\xi_i}(\lambda g_i(z_1) + (1 - \lambda)g_i(z_2)). \tag{4.16}$$

By applying log in the above inequality,

$$\log(F_{\xi_i}(g_i(\lambda z_1 + (1 - \lambda)z_2))) \geq \log(F_{\xi_i}(\lambda g_i(z_1) + (1 - \lambda)g_i(z_2))). \tag{4.17}$$

Functions F_{ξ_i} are $0-$concave by hypothesis. Then

$$\log(F_{\xi_i}(\lambda g_i(z_1) + (1-\lambda)g_i(z_2))) \geq \lambda \log(F_{\xi_i}(g_i(z_1))) + (1-\lambda)\log(F_{\xi_i}(g_i(z_2))). \tag{4.18}$$

By gathering inequality (4.17) and (4.18) we have

$$\begin{aligned} \log(F_{\xi_i}(g_i(\lambda z_1 + (1-\lambda)z_2))) &\geq \log(\lambda F_{\xi_i}(g_i(z_1)) + (1-\lambda)F_{\xi_i}(g_i(z_2))) \\ &\geq \lambda \log(F_{\xi_i}(g_i(z_1))) + (1-\lambda)\log(F_{\xi_i}(g_i(z_2))). \end{aligned} \tag{4.19}$$

The Zhang's Copula evaluated at the point $\lambda z_1 + (1-\lambda)z_2$ is

$$\begin{aligned} \mathbb{C}(F_{\xi_1}(g_1(\lambda z_1 + (1-\lambda)z_2), \ldots, F_{\xi_m}(g_m(\lambda z_1 + (1-\lambda)z_2)) = \prod_{j=1}^{r} \min_{1\leq i\leq m} [F_{\xi_i}(g_i(\lambda z_1 \\ + (1-\lambda)z_2))]^{a_{j,i}}, \end{aligned}$$

where $a_{j,i} \geq 0$.

To simply the notation, $F_{\xi_1}(g_1(\lambda z_1 + (1-\lambda)z_2), \ldots, F_{\xi_m}(g_m(\lambda z_1 + (1-\lambda)z_2)$ is written as $F_\xi(g(\lambda z_1 + (1-\lambda)z_2))$. So,

$$\begin{aligned} \log \mathbb{C}(F_\xi(g(\lambda z_1 + (1-\lambda)z_2))) &= \log\left(\prod_{j=1}^{r} \min_{1\leq i\leq m} [F_{\xi_i}(g_i(\lambda z_1 + (1-\lambda)z_2))]^{a_{j,i}}\right) \\ &= \sum_{j=1}^{r} \log\left(\min_{1\leq i\leq m} [F_{\xi_i}(g_i(\lambda z_1 + (1-\lambda)z_2))]^{a_{j,i}}\right). \end{aligned}$$

As the log function is increasing, $\log \min u = \min \log u$, and therefore

$$\log \mathbb{C}(F_\xi(g(\lambda z_1 + (1-\lambda)z_2))) = \sum_{j=1}^{r} \min_{1\leq i\leq m} \left[\log\left(F_{\xi_i}(g_i(\lambda z_1 + (1-\lambda)z_2)\right)^{a_{j,i}}\right].$$

As $a_{j,i} \geq 0$, the equality becomes

$$\log \mathbb{C}(F_\xi(g(\lambda z_1 + (1-\lambda)z_2))) = \sum_{j=1}^{r} \min_{1\leq i\leq m} \left[a_{j,i} \log\left(F_{\xi_i}(g_i(\lambda z_1 + (1-\lambda)z_2)\right)\right].$$

By using (4.19) in the above equality it becomes

$$\log \mathbb{C}(F_\xi(g(\lambda z_1 + (1-\lambda)z_2))) \geq \sum_{j=1}^{r} \min_{1\leq i\leq m} a_{j,i} \left[\lambda \log(F_{\xi_i}(g_i(z_1))) + (1-\lambda)\log(F_{\xi_i}(g_i(z_2)))\right].$$

The right-most side of the above inequality is greater or equal than

$$
\begin{aligned}
&\sum_{j=1}^{r} \min_{1\le i\le m} a_{j,i}\left[\lambda \log(F_{\xi_i}(g_i(z_1)))\right] + \sum_{j=1}^{r} \min_{1\le i\le m} a_{j,i}\left[(1-\lambda)\log(F_{\xi_i}(g_i(z_2)))\right] \\
&= \lambda \sum_{j=1}^{r} \min_{1\le i\le m} \left[\log(F_{\xi_i}(g_i(z_1)))^{a_{j,i}}\right] + (1-\lambda)\sum_{j=1}^{r} \min_{1\le i\le m} \left[\log(F_{\xi_i}(g_i(z_2)))^{a_{j,i}}\right] \\
&= \lambda \sum_{j=1}^{r} \log\left(\min_{1\le i\le m} (F_{\xi_i}(g_i(z_1)))^{a_{j,i}}\right) + (1-\lambda)\sum_{j=1}^{r} \log\left(\min_{1\le i\le m} (F_{\xi_i}(g_i(z_2)))^{a_{j,i}}\right) \\
&= \lambda \left[\log \prod_{j=1}^{r} \min_{1\le i\le m} (F_{\xi_i}(g_i(z_1)))^{a_{j,i}}\right] + (1-\lambda)\left[\log \prod_{j=1}^{r} \min_{1\le i\le m} (F_{\xi_i}(g_i(z_2)))^{a_{j,i}}\right] \\
&= \lambda \log \mathbb{C}(F_\xi(g(z_1))) + (1-\lambda)\log \mathbb{C}(F_\xi(g(z_2))).
\end{aligned}
$$

It was then demonstrated that

$$
\log \mathbb{C}(F_\xi(g(\lambda z_1 + (1-\lambda)z_2))) \ge \lambda \log \mathbb{C}(F_\xi(g(z_1))) + (1-\lambda)\log \mathbb{C}(F_\xi(g(z_2))),
$$

i.e., the $\log \mathbb{C}(F_{\xi_1}(g_1(z)),\ldots,F_{\xi_m}(g_m(z)))$ is a concave function. In other words, the copula $\mathbb{C}$ is $\alpha-$concave for $\alpha \le 0$. ■

This result is fundamental for this Thesis because this family of copula is log concave and can be used in the problem (4.13). As the assumptions about OA algorithms are assured, the convergence for this class of problem is achieved.

4.3.3 Chance-constrained involving discrete distribution

Until now, we have seen the theory about chance constrained when ξ follows a continuous distribution of probability. In this section we review briefly CCMINLP when ξ follows a discrete distribution of probability.

Consider the chance constrained problem:

$$
\min_{x\in X, x\ge 0} \phi(x) \quad \text{s.t.} \quad P[g(x)\ge \xi] \ge p \tag{4.20}
$$

where $\phi : \mathbb{R}^n \to \mathbb{R}$ is a convex function, $g : \mathbb{R}^n \to \mathbb{R}^m$ is concave function. We assume that the deterministic constraints are expressed by a closed convex set $X \subset \mathbb{R}^n$. The random vector $\xi \in \mathbb{R}^m$ has finite support, that is, there exist vectors $\xi^i \in \mathbb{R}^m, i = 1,\ldots,N$ with $P[\xi = \xi^i] = \pi_i$ for every i where $\pi_i \ge 0$ and $\sum_{i=1}^{N} \pi_i = 1$. We assume without loss of generality that $\xi^i \ge 0$ and $\pi_i \le 1-p$. We can formulate problem (4.20) as a MINLP. To this end, we introduce for each $i = 1, \ldots, N$ a binary variable z_i where

$$z_i = \begin{cases} 0 & \text{if} \quad g(x) \geq \xi^i \\ 1 & \text{if} \quad g(x) \ngeq \xi^i. \end{cases}$$

Then letting $v = g(x)$ we obtain the equivalent problem

$$\begin{aligned} \min_{x,z,v} \quad & \phi(x) \\ \text{s.t.} \quad & g(x) = v \\ & v + \xi^i z_i \geq \xi^i \quad i = 1, \ldots, N \\ & \sum_{i=1}^{N} \pi_i z_i \leq 1 - p \\ & x \geq 0, x \in X, v \geq 0, z \in \{0,1\}^N. \end{aligned} \tag{4.21}$$

Problem (4.21) is a MINLP problem that can be solved by specialized methods.

Another approach to solve chance constrained problems when ξ follows a discrete distribution is using $p-$efficient points.

The $p-$level set of the distribution function $F_\xi(w) = P[\xi \leq w]$ of ξ is defined as

$$\mathcal{Z}_p = \{w \in \mathbb{R}^m | \quad F_\xi(w) \geq p\}.$$

Problem (4.20) can be rewritten as

$$\min_{x \in X} \phi(x) \;\; \text{s.t.} \;\; g(x) \in \mathcal{Z}_p\,. \tag{4.22}$$

It can be proved that for every $p \in (0,1)$ the level set $\mathcal{Z}_p$ is nonempty and closed (see Theorem 4.6.2 and 4.6.3 in [62]).

There exist minimal points in the level set $\mathcal{Z}_p$ with respect to the partial order in $\mathbb{R}^m$ generated by the nonnegative cone $\mathbb{R}^m_+$. These points are called $p-$efficient points.

Definition 4.5. *Let $p \in (0,1)$. A point $v \in \mathbb{R}^m$ is called a $p-$efficient point of the probability distribution F_ξ if $F_\xi(v) \geq p$ and there is not $w \leq v, w \neq v$ such that $F_\xi(w) \geq p$.*

For a given $p \in (0,1)$ let be $\ell = (F^{(-1)}_{\xi_1}(p), \ldots, F^{(-1)}_{\xi_m}(p))$. Then can be proved that for every $w \in \mathbb{R}^m$ such that $F_\xi(v) \geq p$ must satisfy the inequality $w \geq \ell$.

Let be $\mathcal{I}$ an arbitrary index set and let be $v^j, j \in \mathcal{I}$ all $p-$efficient points of ξ. We define the cones

$$K_j = v^j + \mathbb{R}^m_+, j \in \mathcal{I}.$$

By Theorem 4.60 in [62] it follows that $\mathcal{Z}_p = \bigcup_{j \in \mathcal{I}} K_j$. With the results above, problem (4.22) can be rewritten as the following disjunctive semi-infinite formulation

$$\min_{x \in X} \phi(x) \quad \text{s.t.} \quad g(x) \in \bigcup_{j \in \mathcal{I}} K_j \,. \tag{4.23}$$

Denote the convex hull of the $p-$efficient points by E, i.e., $E = \operatorname{conv}\{v^j, j \in \mathcal{I}\}$. Then can be proved that

$$\operatorname{conv}(\mathcal{Z}_p) = E + \mathbb{R}^m_+.$$

Moreover, the set $\operatorname{conv}(\mathcal{Z}_p)$ is nonempty, closed and it is contained in the set of $p-$efficient points. If $\xi \in \mathbb{Z}^m$, Theorem 4.64 of [62] assures that the distribution function F_ξ has finitely many $p-$efficient points. With this assumption, the set $\mathcal{I}$ is a finite set.

Theorem 4.5. *Let $\mathcal{A}$ be the set of all possible values of an integer random vector ξ. If the distribution function F_ξ of ξ is $\alpha-$concave on $\mathcal{A} + \mathbb{Z}^m_+$ for some $\alpha \in [-\infty, \infty]$, then for every $p \in (0,1)$ one has*

$$\mathcal{Z}_p = \{y \in \mathbb{R}^m | y \geq w \geq \sum_{j \in \mathcal{I}} \lambda_j v^j, \sum_{j \in \mathcal{I}} \lambda_j = 1, \lambda_j \geq 0, w \in \mathbb{Z}^m\},$$

where $v^j, j \in \mathcal{I}$ are the $p-$efficient points of F_ξ.

Proof: See Theorem 4.65 in [62]. ■

The consequence of this theorem is that under $\alpha-$concavity assumption, all integer points contained on $\operatorname{conv}(\mathcal{Z}_p) = E + \mathbb{R}^m_+$ satisfy the probability constraint. Under the conditions of Theorem 4.5, problem (4.22) can be formulated as

$$\begin{array}{ll} \min\limits_{x,w,\lambda} & \phi(x) \\ \text{s.t.} & g(x) \geq w \\ & w \geq \sum\limits_{j \in \mathcal{I}} \lambda_j v^j \\ & \sum\limits_{j \in \mathcal{I}} \lambda_j = 1, \\ & \lambda_j \geq 0, j \in \mathcal{I} \quad x \in X, w \in \mathbb{Z}^m. \end{array} \tag{4.24}$$

In problem (4.24), the probability constraint was replaced by algebraic equations and inequality, together with the integrality requirement $w \in \mathbb{Z}^m$. Methods to solve (4.24) require the generation of $p-$efficient points and use an enumeration scheme to identity such points. It is not the scope of this work to study this approach. More about $p-$efficient points can be found in [21, 22, 62]. A recent paper is [67].

Another approach to deal with chance constrained problems with finite support are sample average approximation [47, 48].

The next chapter deals with the task of computing numerically solutions of several CCMINLP problems, some of them having discrete probability distributions, whereas others having continuous distributions approximated or not by copulae.

Chapter 5

Numerical assessment

In this chapter we assess the numerical performance of the proposed OA algorithms on some chance-constrained MINLP problems. In Section 5.1, a hybrid robust/chance-constrained model with finitely many scenarios is considered. The studied model is of great interest in the industry of energy. The main difficulty in this type of problems consists in solving a master subproblem (MILP or MIQP). In Section 5.2 we consider a different application of CCMINLP problems: we investigate a power management planning problem with realistic data. Differently from the application of Section 5.1, the considered chance-constrained problem is based on a continuous probability distribution. As a result, the main difficulty in solving the problem is handling the nonlinear OA's subproblems, rather than the master problem. To overcome this difficulty, we consider an approximation of the problem by replacing the probability by a Copula (which is much easier to evaluate).

5.1 A hybrid robust/chance-constrained model

This section corresponds to Section 5 of paper [20] with different results because a different computer was used. However, the conclusion are similar. We consider the minimization of a linear function $f(x, y) = c_x^\top x + c_y^\top y$ subject to deterministic linear constraints $x \in X$, $y \in Y$, and the stochastic linear constraints

$$A(\omega)x + B(\omega)y \leq \xi \,, \tag{5.1}$$

where $\omega \in \Omega$ and $\xi \in \Xi$ represents different sources of uncertainty. It is important to mention that not all uncertainty are equally well understood. This setting is of interest, for instance, in the industry of energy, where x represents an energy production schedule, y an integer variable modeling importation/investment/"on-off" decisions, and (5.1) means that energy production should meet the energy demand ξ. While the distribution of ξ is very often available (since its characterization has received considerable attention), much less information is available on the uncertainty ω impacting $A(\omega)$ and $B(\omega)$, which are related to the underlying physics of generation plants and/or to the behavior of other generation companies. We follow the lead of [69] and employ a *hybrid robust/chance-constrained* approach to this problem:

$$\begin{cases} \min\limits_{x,y} & c_x^\top x + c_y^\top y \\ \text{s.t.} & \mathbb{P}_\xi[A(\omega)x + B(\omega)y \leq \xi \quad \forall \omega \in \Omega] \geq p \\ & x \in X, \; y \in Y \,, \end{cases} \tag{5.2}$$

where $\mathbb{P}_\xi$ is a probability measure related to the random vector (energy demand) ξ. The joint probabilistic constraint in (5.2) requires that all stochastic inequalities hold simultaneously with high enough probability $p \in (0, 1]$. When every row $a_i(\omega)$ and $b_i(\omega)$ of matrices $A(\omega)$ and $B(\omega)$ depend on the random vector ω in the form

$$a_i(\omega)^\top x + b_i(\omega)^\top y = \bar{a}_i^\top x + \bar{b}_i^\top y + \langle P_i \omega, (x, y) \rangle,$$

with given $\bar{a}_i \in \mathbb{R}^{n_x}$, $\bar{b}_i \in \mathbb{R}^{n_y}$, $P_i \in \mathbb{R}^{(n_x+n_y)\times n_i}$ and $\omega \in \Omega_i := \{\omega \in \mathbb{R}^{n_i} : \|\omega\| \leq \kappa_i\}$ (with given $\kappa_i > 0$), the well-established theory of robust optimization [6] applies and the above problem can be rewritten in the following equivalent formulation

$$\begin{cases} \min\limits_{x,y} & c_x^\top x + c_y^\top y \\ \text{s.t.} & \mathbb{P}_\xi[\bar{a}_i^\top x + \bar{b}_i^\top y + \kappa_i \|P_i^\top (x, y)\| \leq \xi_i \quad \forall i = 1, \ldots, m] \geq p \\ & x \in X, \; y \in Y \,. \end{cases}$$

We suppose that ξ takes values in a finite set $\Xi = \{\xi^s, s \in S\} \subseteq \mathbb{R}^m$ of possible realizations with associated weights $\pi_s > 0$ with $\sum\limits_{s\in S} \pi_s = 1$. Under this assumption, a binary variable $z_s \in \{0, 1\}$ for each $s \in S$ is introduced which dictates whether or not $\bar{a}_i^\top x + \bar{b}_i^\top y + \kappa_i \|P_i^\top (x, y)\| \leq \xi_i^s$ is satisfied for all i. By using a "big M" formulation, with $M > 0$ a given parameter, the problem of interest can be reformulated as

$$\begin{cases} \min\limits_{x,y,z} & c_x^\top x + c_y^\top y \\ \text{s.t.} & \max\limits_{s\in S}\{\bar{a}_i^\top x + \bar{b}_i^\top y + \kappa_i \|P_i^\top (x, y)\| - \xi_i - M_i^s z_s\} \leq 0, \quad i = 1, \ldots, m \\ & \sum\limits_{s\in S} \pi_s z_s \leq 1 - p \\ & x \in X, \; y \in Y, \; z_s \in \{0, 1\}. \end{cases} \tag{5.3}$$

Problem (5.3) fits the general formulation (3.1) and hence can be solved by variants of the Outer Approximation Algorithm 3.1.

In spite of the nonlinear constraints in problem (5.3), which are nonsmooth ones, a second order constrained formulation for problem (5.3) can be obtained by introducing auxiliary variables w_i, $i = 1, \dots, m$, and additional constraints

$$\left\{\begin{array}{ll} \min\limits_{x,y,z,w} & c_x^\top x + c_y^\top y \\ \text{s.t.} & \bar{a}_i^\top x + \bar{b}_i^\top y + \kappa_i w_i - \xi_i^s \le M_i^s z_s \quad i = 1, \dots, m \quad \text{and all} \quad s \in S \\ & (x,y)^\top (P_i P_i^\top)(x,y) \le w_i^2 \quad i = 1, \dots, m \\ & \sum\limits_{s \in S} \pi_s z_s \le 1 - p \\ & x \in X,\ y \in Y,\ z_s \in \{0,1\},\ w \ge 0. \end{array}\right. \tag{5.4}$$

This formulation, denoted by "*Monolithic*", replaces m nonsmooth constraints with $m|S|$ linear and m conic constraints. Thus, the state-of-the-art mixed-integer second-order algorithms can be applied to solve the above problem and compare the numerical performance with `OA` algorithms applied to (5.3). In this numerical tests, the Monolithic formulation is solved by `Gurobi` [34].

5.1.1 Test problems, solvers and results

The integer set Y in the problem (5.3) was set as $0 \le y_i \le 3$, $y_1 + y_2 \ge 3$ and $\sum\limits_{i=1}^{5} y_i \le 5$. Instances for problem (5.3) are generated following the procedure: First, the dimension of problem is set as $n_y = 5$, $n_x \in \{30 - n_y, 40 - n_y\}$, $m \in \{15, 20, 25\}$ and the number of scenarios $N \in \{30, 40, 50, 60\}$. Next, $n_i = n_x + n_y$, $\kappa_i = \frac{1}{2}$ for all $i = 1, \dots, m$, and $p \in \{0.9, 0.95\}$. The probability of every scenario s is $\pi_s = \frac{1}{N}$. The vectors $\bar{a}_i \in \mathbb{R}^{n_x}, \bar{b}_i \in \mathbb{R}^{n_y}$ were generated with entries uniformly distribution on the interval $[0, 10]$. The vector cost $c = (c_x, c_y)$ was generated with entries on the interval $[-100, 0)$ and the matrix P_i with coefficients following a normal distribution $N(0,1)$. An initial point (x_0, y_0) was chosen which is feasible for construct and the scenarios was set as $\xi_i = \bar{a}_i^\top x_0 + \bar{b}_i^\top y_0 + \kappa_i \|P_i^\top (x_0, y_0)\| + \bar{\xi}_i$ where $\bar{\xi}_i$ is uniformly on the interval $[0, 100]$. The constants M_i^s are the same for all scenarios and was chosen with an ad-hoc approach. In this problem, the number of continuous, integer and binary variables are respectively n_x, n_y and N. For each one of this configuration, two instances are generated by changing the seed of the pseudorandom number generator. In a total, $2 \cdot 3 \cdot 4 \cdot 2 \cdot 2 = 96$ different test problems were considered.

Solvers. These nonsmooth convex mixed-integer programs are solved with the following solvers, coded in/or called from MATLAB version 2017a:

- `Monolithic`: solver `Gurobi` applied to the mixed-integer quadratically constrained programming problem (5.4);

- `OA`: it is an implementation of the outer-approximation Algorithm 3.1 with $\mu_k = 0$ for all k (the classic algorithm), where in Step 1 the (nonsmooth convex) nonlinear subproblems are solved by the Bundle Algorithm 2.1. In this solver, the MILP subproblem (3.19) is solved by `Gurobi` to define the next integer iterate;
- `OA-1pt`: as solver `OA`, with the difference that trial integer iterates are defined by applying `Gurobi` to the MILP subproblem (3.19) and halting the solver as soon as a feasible point is found;
- $\mathtt{OA}_1$: as solver `OA`, with integer iterates defined by solving the regularized MILP subproblem (3.19) with the ℓ_1 norm, i.e., $\|\cdot\|_\diamond = \|\cdot\|_1$. The stability center was set as the current iterate and the prox parameter as $\mu_k = 10$ for all k;
- $\mathtt{OA}_\infty$: as solver $\mathtt{OA}_1$, with the ℓ_1 norm replaced by ℓ_∞;
- $\mathtt{OA}_2$: as solver $\mathtt{OA}_1$, with the ℓ_1 norm replaced by ℓ_2. In this case, the subproblem defining the next iterate is no longer a MILP, but a MIQP;
- `ECPM`: this is an implementation of the *extended cutting plane method* of [76]. It is the same solver employed in [16];
- `ELBM`: this is the *extended level bundle method* of [16] with the current iterate rule to define the stability center and ℓ_1-norm for the stability function.

All the solvers employed a relative stopping test with tolerance 10^{-3}, and time limit (for solving each problem) of 3600 seconds. Numerical experiments were performed on a computer with Intel(R) Core(TM), i7-5500U, CPU @ 2.40 GHz, 8G (RAM), under Windows 10, 64Bits.

Numerical experiments. The performance profiles [23] of the eight considered solvers on the 96 instances of problem (5.3) are presented in Figure 5.1. Given a set of problems P and a set of methods S it is possible to compare the performance of these methods on problems using any metric, for example, CPU time. For each problem $p \in P$ and solver $s \in S$, a measure is defined

$$t_s(p) = \text{CPU time request to solve problem p by solver s.}$$

Next, a best achieve is also defined as, $t_s^*(p) = \min_{s \in S} t_s(p)$ (the best CPU time to solve problem p). For each solver $s \in S$, the comparison of its performance in solving a problem $p \in P$ in relation to the best method is done using the performance ratio given by

$$r_s(p) = \frac{t_s(p)}{t_s^*(p)}. \tag{5.5}$$

If the method s fails to solve problem p, then $t_s(p) = \infty$. The performance ratio shows the behavior of a method in solving a given problem. For a more general view the performance profile of method $s \in S$ is defined as

$$\rho_s(\gamma) := \frac{\text{number of problems } p \text{ such that } r_s(p) \leq \gamma}{\text{total number of problems}}. \tag{5.6}$$

By replacing (5.5) in (5.6) an equivalent form is obtained

$$\rho_s(\gamma) := \frac{\text{number of problems } p \text{ such that } t_s(p) \leq \gamma\, t^*(p)}{\text{total number of problems}}.$$

These numbers give the proportion of problems solved by solver s within a factor γ. Therefore, the value $\rho_s(1)$ gives the probability of the solver s to be the best by a given criteria. Furthermore, unless $t_s(p) = \infty$ (which means that solver s failed to solve problem p), it follows that $\lim_{\gamma \to \infty} \rho_s(\gamma) = 1$. Thus, the higher is the line, the better is the solver. The image at the top of Figure 5.1 corresponds to the performance profiles of CPU time required by the methods on all the instances.

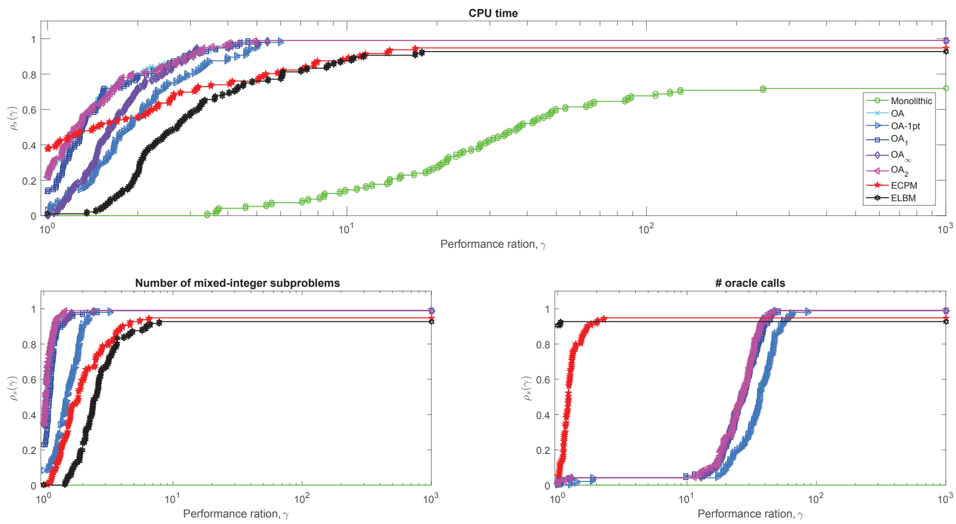

Figure 5.1: Performance profile on 96 problems (logarithmic scale): CPU time, oracle's calls and number of solved mixed-integer subproblems.

Overall, solver $\texttt{OA}_2$ was the most robust with respect to CPU time, followed by `OA`, $\texttt{OA}_1$ and $\texttt{OA}_\infty$. Although solver `ECPM` failed to solve 5 out of 96 problems in less than one hour, `ECPM` was the fastest method in 37% of the problems, followed by `OA` and $\texttt{OA}_2$(both with approximately 22%) and $\texttt{OA}_1$ (14%). As shown by Figure 5.1, solver `Monolithic` was not competitive in these instances (except in the small ones).

The left-bottom image in Figure 5.1 reports the solvers performance with respect to the number of master subproblems (MILP or MIQP) solved by each method (except `Monolithic`). Concerning this attribute, solver $\texttt{OA}_2$ was the most efficient one: overall, this method required solving less master subproblems and, as a consequence, less nonlinear subproblems (that are not trivial to solve, since they are nonsmooth programs). Solver $\texttt{OA}_1$ also provided a good performance on the number of (MILP) subproblems.

Finally, the right-bottom image in Figure 5.1 presents performance profiles with respect to the number of oracle calls. Concerning this attribute, ELBM and ECPM provided (for the solved instances) better performances than the OA solvers, corroborating in this manner with [16]: when functions are costly (which is not the case in the considered problem), cutting-plane methods as ELBM and ECPM seem to be so far the methods of choice. However, these solvers could not solve some of the instances in one hour of time limit.

The information provided by Figure 5.1 are complemented with two tables. Table 5.1 reports (for all the eight solvers) the number of problems that could not be solved within one hour of processing.

Table 5.1: Number (and percentage) of problems that could not be solved in the time limit of one hour.

Solver	Number of fails	Percentage
Monolithic	27	28.12%
OA	1	1.04%
OA-1pt	1	1.04%
$\texttt{OA}_1$	1	1.04%
$\texttt{OA}_\infty$	1	1.04%
$\texttt{OA}_2$	1	1.04%
ECPM	5	5.21%
ELBM	7	7.29%

All OA solvers were able to solve the majority of the problems, while the other solvers could not solve the larger ones. CPU time (in seconds) of every solver on every problem's instance are presented in Table 5.2. The first column corresponds to the seed used to generate the problem instances, and columns 2 up to 5 report the problem's dimension. The total CPU time in hours is reported on the last line of the table.

Table 5.2: CPU time required by the eight solvers on all the ninety six test problems. The asterisk stands for unsolved problem (within one hour).

Problem's data					Solvers							
seed	m	N	p	n	Monolithic	OA	OA-1pt	$\texttt{OA}_1$	$\texttt{OA}_\infty$	$\texttt{OA}_2$	ECPM	ELBM
0	15	30	0.90	30	411.89	13.84	19.92	14.74	16.27	21.92	14.67	24.50
0	15	30	0.95	30	7.99	6.03	10.80	6.40	5.55	6.13	2.14	5.60
0	15	40	0.90	30	1093.32	23.27	54.10	31.01	49.03	27.01	42.46	78.04
0	15	40	0.95	30	218.44	8.44	21.85	9.36	9.66	10.68	6.45	15.06
0	15	50	0.90	30	933.01	47.79	61.86	39.08	95.73	46.54	102.46	169.22
0	15	50	0.95	30	388.12	8.53	21.23	9.53	10.28	7.62	4.46	9.33
0	15	60	0.90	30	3600.45*	84.87	108.99	108.74	157.94	125.90	605.06	894.03
0	15	60	0.95	30	487.35	18.47	38.60	19.59	21.21	15.88	22.64	33.87
0	20	30	0.90	30	377.46	22.45	40.02	26.70	25.29	22.86	17.50	26.86
0	20	30	0.95	30	29.70	11.14	13.80	12.87	14.94	13.49	2.75	4.89
0	20	40	0.90	30	1940.36	51.06	49.35	78.89	61.54	42.87	95.18	123.81
0	20	40	0.95	30	264.78	19.46	24.07	22.48	19.85	19.83	11.46	15.51
0	20	50	0.90	30	3600.55*	121.38	123.45	133.28	168.73	136.26	615.62	661.36
0	20	50	0.95	30	365.46	20.42	27.46	25.74	24.58	26.15	9.82	19.88
0	20	60	0.90	30	3600.61*	378.85	387.21	248.87	369.35	398.53	2516.52	3607.79*
0	20	60	0.95	30	1747.97	61.90	48.60	49.63	55.56	41.86	61.75	84.56

0	25	30	0.90	30	1873.46	30.10	28.33	27.12	43.09	24.62	21.08	34.21
0	25	30	0.95	30	67.48	13.29	19.33	13.90	13.38	13.06	3.23	12.37
0	25	40	0.90	30	3173.34	25.17	31.77	19.80	33.45	17.68	12.92	26.62
0	25	40	0.95	30	266.32	27.47	34.25	25.02	24.97	26.95	19.15	38.95
0	25	50	0.90	30	3600.66*	61.28	43.35	55.64	64.07	37.25	96.00	226.40
0	25	50	0.95	30	435.86	22.93	25.88	22.90	26.10	25.96	15.63	33.67
0	25	60	0.90	30	3600.74*	54.43	82.01	81.81	115.40	70.06	432.93	490.82
0	25	60	0.95	30	829.24	28.51	39.14	29.82	33.72	28.82	35.04	55.17
0	15	30	0.90	40	825.62	40.40	83.13	43.22	49.23	45.53	61.23	77.31
0	15	30	0.95	40	29.76	21.64	28.09	19.60	21.03	21.92	8.72	16.60
0	15	40	0.90	40	2539.62	79.51	112.78	85.20	86.42	79.90	202.85	260.76
0	15	40	0.95	40	285.58	35.14	72.53	39.49	42.11	46.35	49.43	60.76
0	15	50	0.90	40	3600.37*	148.28	249.20	110.73	160.70	131.77	591.12	673.54
0	15	50	0.95	40	391.90	32.12	52.67	31.50	35.69	36.83	20.81	46.63
0	15	60	0.90	40	3600.42*	258.02	223.97	330.01	329.30	294.80	2582.24	3605.11*
0	15	60	0.95	40	1037.53	57.33	97.85	63.53	62.02	56.62	91.18	139.36
0	20	30	0.90	40	340.55	42.68	83.13	49.61	44.49	44.05	24.96	40.32
0	20	30	0.95	40	46.08	24.88	37.68	23.66	24.32	25.30	8.23	11.94
0	20	40	0.90	40	3386.89	63.38	97.95	63.59	82.44	53.80	112.69	129.37
0	20	40	0.95	40	130.57	37.49	65.94	36.38	35.86	37.38	33.06	28.51
0	20	50	0.90	40	3600.57*	80.74	150.94	108.63	107.54	84.98	282.99	365.57
0	20	50	0.95	40	510.08	37.56	67.19	38.60	36.15	40.92	13.50	27.08
0	20	60	0.90	40	3600.64*	246.19	229.49	259.83	255.72	225.18	1753.08	1946.67
0	20	60	0.95	40	499.31	61.92	81.34	54.94	69.30	54.71	35.96	75.96
0	25	30	0.90	40	3600.65*	281.92	449.04	260.59	319.36	216.12	2942.72	2054.70
0	25	30	0.95	40	579.42	29.93	64.82	32.05	30.16	41.08	76.23	62.50
0	25	40	0.90	40	3600.75*	586.78	727.09	519.04	620.05	722.95	3600.10	3605.11
0	25	40	0.95	40	3600.73*	96.38	165.87	72.72	78.52	89.02	428.04	311.99
0	25	50	0.90	40	3600.85*	686.96	965.57	898.29	1096.53	1074.16	3600.04*	3605.07*
0	25	50	0.95	40	3600.91*	84.06	161.17	103.57	99.37	98.79	612.98	378.62
0	25	60	0.90	40	3600.96*	3600.64*	3601.81*	3630.99*	3601.47*	3600.84*	3600.05*	3605.08*
0	25	60	0.95	40	3600.96*	243.48	274.16	269.38	440.80	238.34	1855.23	2030.06
1	15	30	0.90	30	403.92	13.72	18.29	12.39	23.23	10.43	11.48	25.33
1	15	30	0.95	30	36.13	5.23	10.36	4.53	6.97	5.61	2.42	7.76
1	15	40	0.90	30	1383.36	24.41	56.29	16.29	32.31	16.73	32.65	58.79
1	15	40	0.95	30	185.04	8.82	17.45	8.46	12.40	10.12	9.18	23.25
1	15	50	0.90	30	1560.95	44.82	55.54	38.98	81.42	29.05	89.88	213.77
1	15	50	0.95	30	170.54	8.39	19.06	9.53	10.20	9.32	6.56	18.12
1	15	60	0.90	30	3600.32*	71.45	136.74	95.85	121.39	85.24	372.76	436.56
1	15	60	0.95	30	713.57	15.35	21.79	14.89	21.75	18.26	25.44	42.55
1	20	30	0.90	30	188.75	13.08	15.78	12.08	15.31	9.54	9.39	17.57
1	20	30	0.95	30	13.86	8.24	5.92	4.77	5.48	5.35	1.87	3.61
1	20	40	0.90	30	684.83	22.76	24.26	20.47	30.05	22.83	48.13	66.07
1	20	40	0.95	30	159.82	10.73	13.11	12.05	10.41	10.85	3.30	8.91
1	20	50	0.90	30	3484.95	29.04	31.61	29.05	46.14	29.66	63.28	91.53
1	20	50	0.95	30	251.40	10.22	14.99	12.17	11.66	12.00	3.77	11.55
1	20	60	0.90	30	3600.60*	59.76	50.54	49.10	83.15	63.56	156.22	178.26
1	20	60	0.95	30	287.24	14.23	21.66	18.82	19.90	10.45	13.38	21.14
1	25	30	0.90	30	503.61	13.77	19.24	15.59	18.28	13.88	13.87	20.68
1	25	30	0.95	30	52.13	12.41	10.28	13.69	12.53	12.30	3.12	7.94
1	25	40	0.90	30	2356.79	21.26	30.16	18.11	25.40	20.14	47.75	41.55
1	25	40	0.95	30	285.75	18.32	19.63	17.44	21.12	19.63	11.68	18.27
1	25	50	0.90	30	2637.16	24.62	38.56	33.36	44.60	28.04	101.90	148.34
1	25	50	0.95	30	141.11	13.11	14.84	13.34	17.03	16.34	9.32	18.56
1	25	60	0.90	30	3600.96*	42.29	53.30	32.04	50.99	39.51	353.11	331.32
1	25	60	0.95	30	1191.08	23.95	25.77	26.80	34.62	26.94	55.93	71.24
1	15	30	0.90	40	98.96	28.14	36.39	28.76	27.82	28.33	11.83	23.18
1	15	30	0.95	40	16.09	15.32	12.13	14.97	13.68	14.54	4.41	10.03
1	15	40	0.90	40	358.24	37.94	39.30	33.26	41.16	37.48	29.50	46.69
1	15	40	0.95	40	38.50	20.95	19.85	23.60	20.85	24.65	10.46	19.36
1	15	50	0.90	40	531.15	59.80	82.84	61.33	63.00	56.12	128.66	168.15
1	15	50	0.95	40	104.15	26.14	30.77	30.87	26.06	26.29	12.40	33.70
1	15	60	0.90	40	755.10	78.43	99.32	86.41	108.60	21.64	300.95	365.36
1	15	60	0.95	40	141.04	38.43	50.43	45.97	43.47	42.52	26.26	57.31
1	20	30	0.90	40	1141.73	61.03	62.43	62.96	64.45	52.91	61.86	88.89
1	20	30	0.95	40	120.23	27.00	32.59	33.56	31.95	32.81	12.19	30.27
1	20	40	0.90	40	3600.77*	122.07	170.08	118.00	195.61	122.81	594.95	570.18
1	20	40	0.95	40	1071.87	49.75	73.90	50.79	49.94	25.40	70.67	83.14
1	20	50	0.90	40	3601.05*	162.02	196.93	176.61	208.95	135.69	2288.86	1541.48
1	20	50	0.95	40	576.97	49.89	101.12	58.72	52.48	52.96	53.42	84.75
1	20	60	0.90	40	3600.96*	214.88	355.88	246.94	416.04	410.33	3600.08*	3605.08*
1	20	60	0.95	40	1589.51	65.96	125.11	73.37	78.31	57.07	231.53	230.40
1	25	30	0.90	40	3068.97	49.86	78.35	59.33	64.42	47.13	101.36	133.00
1	25	30	0.95	40	53.80	22.94	35.30	19.32	22.68	26.91	7.60	14.89
1	25	40	0.90	40	3600.89*	137.77	155.86	123.90	183.12	96.60	593.01	677.54
1	25	40	0.95	40	1614.16	37.07	64.11	40.20	42.12	38.81	47.03	75.26
1	25	50	0.90	40	3600.96*	186.12	129.90	172.66	193.90	199.77	1258.34	2310.61
1	25	50	0.95	40	3601.06*	29.94	42.31	32.72	36.38	32.99	29.37	60.29
1	25	60	0.90	40	3601.09*	366.88	365.36	562.11	411.18	691.30	3600.05*	3605.11*
1	25	60	0.95	40	3601.54*	85.98	110.02	83.73	101.60	63.68	200.20	268.61
Total CPU time in hours					41.9 h	2.9 h	3.5 h	3.1 h	3.5 h	3.2 h	11.7 h	12.7 h

Solvers $\texttt{OA}$ and $\texttt{OA}_1$ were faster, followed by $\texttt{OA}_2$. The solver $\texttt{OA}_2$ solve a MIQP per iteration which is more expensive than MILP subproblem. The outer-approximation algorithms were 4 times faster than the extended cutting-plane methods.

Note that the numerical performances of the regularized $\texttt{OA}$ algorithms are very close to each other. Better performances are expected to be obtained if μ_k is iteratively updated.

5.2 A power system management problem

Consider a power management model consisting of a hydro power plant and a wind farm. Electricity that is generated by both units has two purposes: first attend the local community power demand and secondly the leftover is sold on the market. The energy that is generated by the wind farm is designated to supply the local community demand only. If it is not enough then the remaining demand is covered by the hydro power plant. The residual energy portion generated by the hydro power plant is then sold to the market with the aim of maximizing the profit, which varies according to the given energy price. Since the intention is to consider a short time planning period (e.g. one day) the assumption is that the only uncertainty in this energy planning problem comes from the wind power production. As a result the approach will consider the inflow to the hydro plant, market prices and energy demand as known parameters. The hydro plant counts with a reservoir that can be used to store water and adapt the water release strategy to better achieve profit according the price versus demand: the price of electricity varies during the day, thus it is convenient to store water (if possible) to generate electricity at moments of the day deemed more profitable.

In order to exclude production strategy that can be optimum in a short period of time and can harm the near future energy supply (e.g. the planner can be willing to use all water in the reservoir to produce energy to maximize profit because the energy prices are higher and in the next hour there is no enough water to produce energy in case the wind farm is failing to supply the local community leading to a blackouts), a level constraint is imposed for the final water level in the hydro reservoir i.e. it cannot be lower of a certain level l^*.

The decision variables of the problem are the leftover energy to supply the local community and the residual energy to be sold to the market (both generated by the hydro power plant). Since the main purpose of the problem is to maximize the profit for the power plant owner then the objective function is profit maximization. Some of the constraints of this problem are simple bounds of water release which are given by the operational limits of the turbine (usually provided per the manufacturer), lower and upper bounds of hydro reservoir filling level and demand satisfaction. As in the paper [3], the demand satisfaction constraint will be dealt with in a probabilistic manner: random constraints in which a decision has to be taken prior to the observation of the random variable are not well-defined in the context of an optimization problem. This motivates the formulation of a corresponding probabilistic constraint in which a decision is defined to be feasible if the underlying random constraint is satisfied under this decision at least with a certain specified probability p.

A further characteristic of this model is to consider binary decision variables. These variables are needed because turbines cannot be operated using an arbitrary level: they are either off or on (working in a positive level). Such on/off constraints are easily modeled by binary variables. By discretizing the time horizon (one day) into T intervals (hours), the resulting optimization problem is described below:

$$
\begin{aligned}
\max_{x,y,z} \quad & \sum_{t=1}^{T} \pi_t z_t \\
\text{s.t.} \quad & P[x_t + \xi_t \geq d_t \quad \forall t = 1, \ldots, T] \geq p \\
& y_t \underline{v} \leq x_t + z_t \leq y_t \bar{v} && \forall t = 1, \ldots, T \\
& x_t, z_t \geq 0 && \forall t = 1, \ldots, T \\
& y_t \in \{0, 1\} && \forall t = 1, \ldots, T \\
& \underline{l} \leq l_0 + t\omega - \tfrac{1}{\chi} \sum_{\tau=1}^{t} (x_\tau + z_\tau) \leq \bar{l} && \forall t = 1, \ldots, T \\
& l_0 + T\omega - \tfrac{1}{\chi} \sum_{\tau=1}^{t} (x_\tau + z_\tau) \geq l^*,
\end{aligned}
\tag{5.7}
$$

where

- z_t is the residual energy which is produced by the hydro power plant in time interval t that is sold to market;
- π_t is the energy price in the time t;
- x_t is the amount of energy generated by hydro power plant to supply the remaining demand on local community on time t;
- d_t is the local community demand on time t, which is assumed to be known (due to the short planning horizon of one day);
- ξ_t is the random energy generated by the wind farm on time t;
- P is the probability measure associated to random vector ξ. As in [3], we assume that the wind power generation follows a multivariate normal distribution with mean vector μ and a positive definite correlation matrix Σ. This assumption leads to this function to be differentiable, see Theorem A.3 below;

- $p \in (0, 1]$ is the given parameter to ensure confidence level for the demand satisfaction;
- $\underline{v}$ and $\bar{v}$ are the lower and upper respectively operations limits of the hydro power plant turbine;
- y_t is the binary variable modeling turbine turn on/turn off;
- l_0 is the initial water level of the hydro power plant reservoir at the beginning of the horizon;
- $\underline{l}$ and $\bar{l}$ are the lower and upper water levels respectively in the hydro power plant reservoir at any time;
- ω denotes the constant amount of water inflow to the hydro power plant reservoir at any time t;
- χ represents a conversion factor between the released water and the energy produced by the turbine: one unit of water released corresponds to χ units of energy generated;
- l^* is the minimum level of water into the hydro power plant reservoir in the last period T of the time horizon.

The difficulty of this problem consists in dealing with hard chance constraint (even though it is a differentiable function) and the binary variables. As in the previous section, this problem will be solved using variants of the `OA` algorithm and two more methods to compare the results. The methods are `ECPM` and `ELBM`. It is important to observe that in this problem the binary variables are not present at the probability constraint, however on chance constraint these variables impact the continuous variables x_t.

As in [3], we assume that the wind power generation follows a multivariate normal distribution[1] with mean vector μ and positive definite covariance matrix Σ. We can replace the first inequality in problem (5.7) by a equivalent one. To this end, consider the following development:

$$
\begin{aligned}
P[x_t + \xi_t \geq d_t \quad \forall t = 1, \ldots, T) &= P(x + \xi \geq d] \\
&= P[\xi \geq d - x] \\
&= P[-\xi \leq x - d] \\
&= F_{-\xi}(x - d).
\end{aligned}
\tag{5.8}
$$

Using the results of equality (5.8) and the results of $0-$concavity by Prékopa (Theorem 4.2) the following function is convex:

$$
f_1(x, y) = \log p - \log F_{-\xi}(x - d), \tag{5.9}
$$

[1] See Appendix A.

where $x = (x_1, x_2, \ldots, x_T)$ and $d = (d_1, \ldots, d_T)$. By replacing the function $-P[x_t + \xi_t \geq d_t \quad \forall t = 1, \ldots, T] + p$ in the first inequality in (5.7) with f_1 above the resulting (equivalent) problem fits the general formulation (3.1), i.e., a convex MINLP.

5.2.1 Problem's data

In this Thesis we will solve a similar problem as [3] but with different data[2]. Problem (5.7) couples one wind farm with one hydro power plant to supply energy to one city (or region). The remain of the energy is sold at the market. The demand considered in this problem was extracted from the `ONS` website (`www.ons.org.br`). `ONS` is the Brazilian independent system operator. The behavior of Brazilian demand of energy in all days of the week can be seen in Figure 5.2 below. As expected, the demand of energy is higher on working days than in on weekends and the demand peak is reached around `7pm` daily.

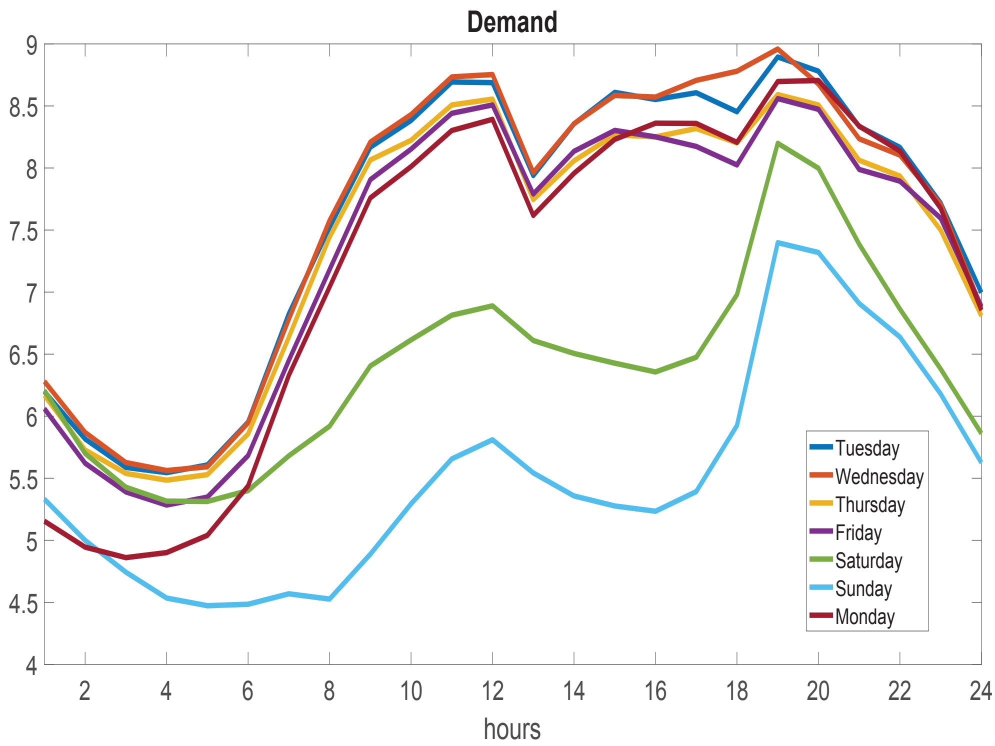

Figure 5.2: Demand

In our numerical tests, the considered daily demands corresponds to eighty percent of averaged demand Southern region of Brazil, divided by the number of cities in such a region. Table 5.3 shows the total demand for each hour in the South of Brazil along the first week of August of year 2017. In this part of Brazil, there exist 1191 cities, so the demand d_t was set as $d_t = \frac{0.8 \cdot D_t}{1191}$, where D_t is the real demand from Table 5.3. Disclaimer, the time on Table 5.3 start counting from midnight. Note that we have only one city (or

[2]We did not have access to the data used in [3].

region) and two sources of energy: hydro power plant and wind farm. The price π_t of energy is directly proportional to the demand and varies between 166.35 and 266.85 by $MW\backslash$hour.

Table 5.3: Total demand by day in MegaWatts. [3]

	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday
1	9227.99	9350.23	9182.76	9022.94	9241.86	7937.74	7673.28
2	8658.61	8737.46	8526.88	8367.84	8485.47	7442.99	7361.51
3	8317.70	8376.20	8246.14	8024.16	8079.97	7061.61	7235.43
4	8255.23	8280.63	8165.01	7866.53	7913.52	6750.67	7296.04
5	8347.14	8325.56	8230.96	7961.15	7909.38	6658.99	7502.61
6	8861.46	8852.44	8718.87	8459.56	8040.18	6676.26	8095.74
7	10160.85	10096.96	9879.40	9601.52	8457.01	6801.74	9422.43
8	11196.35	11280.72	11080.43	10689.89	8811.82	6738.44	10482.32
9	12158.75	12220.84	12006.24	11770.01	9534.67	7276.68	11550.99
10	12480.01	12551.32	12242.64	12136.07	9847.49	7880.52	11926.67
11	12941.58	13003.01	12666.33	12566.19	10143.86	8420.08	12359.81
12	12934.24	13031.27	12736.43	12666.51	10256.33	8649.89	12494.15
13	11819.08	11854.94	11530.89	11597.79	9840.34	8251.92	11340.65
14	12439.62	12442.20	11996.27	12110.67	9685.11	7975.58	11846.42
15	12817.28	12781.02	12308.21	12361.02	9568.39	7855.23	12253.40
16	12731.74	12761.31	12283.81	12283.77	9463.17	7792.09	12447.63
17	12812.86	12961.39	12383.91	12168.82	9639.17	8027.79	12444.58
18	12584.67	13071.17	12210.86	11946.67	10389.63	8820.52	12217.66
19	13242.21	13337.65	12790.96	12742.76	12206.38	11014.10	12947.36
20	13071.51	12916.61	12665.62	12614.07	11907.12	10896.47	12960.04
21	12408.24	12259.99	12002.74	11892.67	10995.60	10285.13	12412.71
22	12157.55	12064.61	11812.95	11752.07	10217.42	9882.17	12112.37
23	11489.58	11473.23	11164.35	11307.19	9498.72	9199.07	11426.03
24	10413.51	10233.62	10134.66	10249.81	8724.79	8374.14	10203.60

The configuration of hydro power plant reservoir is mirrored from [3] and in this problem is set as $\underline{l} = 5000\ hm^3$ (cubic hectometre), $\bar{l} = 10000\ hm^3$ and $l_0 = l^* = 7500\ hm^3$. The amount of water inflow is a constant $\omega = 2500\ hm^3$ each hour and the conversion factor is $\chi = 0.0035 MWm^3$. When the turbines are turned on, the minimum power generation is 5 megawatts per hour and the maximum generation is 20 megawatts per hour.

As previously mentioned, the random variable $\xi = (\xi_1, \dots, \xi_T)$ of wind farm generation follows the distribution $N(\mu, \Sigma)$, where μ denotes the vector of expected values of ξ and Σ is the covariance matrix associated with the components of ξ. The constant mean vector as set as $\mu_i = 1.1 * \min\limits_{1 \le j \le T} d_j\ \forall i = 1, \dots, T$ and

$$\Sigma_{ij} = \begin{cases} \frac{1}{16} & \text{if} \quad i = j \\ \frac{17}{320} & \text{if} \quad i \neq j. \end{cases}$$

These assumptions assures that function (A.1) is differentiable (Σ has full rank)[4] and the

[3]This data corresponds to the first week of August 2017.

[4]We recall that differentiability is not required by our solvers.

coordinates $\xi_1, \xi_2, \ldots, \xi_m$ are dependent.

5.2.2 Problem's approximation

In this section we replace function (5.9) with a Zhang's Copula (4.14) in problem (5.7):

$$
\begin{array}{lll}
\max\limits_{x,y,z} & \sum\limits_{t=1}^{T} \pi_t z_t & \\
\text{s.t.} & \log p - \log(\mathbb{C}(F_{-\xi_1}(x_1 - d_1), \ldots, F_{-\xi_T}(x_T - d_T)) \leq 0 & \\
& y_t \underline{v} \leq x_t + z_t \leq y_t \bar{v} & \forall t = 1, \ldots, T \\
& x_t, z_t \geq 0 & \forall t = 1, \ldots, T \\
& y_t \in \{0, 1\} & \forall t = 1, \ldots, T \\
& \underline{l} \leq l_0 + t\omega - \frac{1}{\chi} \sum\limits_{\tau=1}^{t} (x_\tau + z_\tau) \leq \bar{l} & \forall t = 1, \ldots, T \\
& l_0 + T\omega - \frac{1}{\chi} \sum\limits_{\tau=1}^{t} (x_\tau + z_\tau) \geq l^*, &
\end{array}
\tag{5.10}
$$

where $F_{-\xi_i}$ are the margins associate to distribution function $F_{-\xi}$.

As $F_{-\xi_i}, i = 1, \ldots, m$ are $0-$concave (because $-\xi_i \sim N(-\mu_i, \sigma_i^2)$) and

$$g(x, y) = (x_1 - d_1, x_2 - d_2, \ldots, x_T - d_T)$$

is concave, by Preposition 4.1

$$\mathbb{C}(F_{-\xi_1}(x_1 - d_1), \ldots, F_{-\xi_T}(x_T - d_T)$$

is $0-$concave and hence

$$f_1(x, y) = \log p - \log(\mathbb{C}(F_{-\xi_1}(x_1 - d_1), \ldots, F_{-\xi_T}(x_T - d_T))$$

is convex. Consequently, problem (5.10) is a nonsmooth convex MINLP.

The only difference between problems (5.7) and (5.10) is the first constraint. We will see in the next section that problem (5.7) is very challenging computationally. Instead of solving problem (5.7) we will get an approximate solution by solving problem (5.10). Such approximate solution will be a feasible point of (5.7) if it satisfy the chance constraint of problem (5.7).

Parameters of Zhang's Copula

One of difficulties in using copulae is to find its coefficients that model with accuracy the probability constraint. The parameters of Zhang's Copula depend on the size of the problem. If the random vector ξ has dimension T then the number of parameters is $1+rT$:

$$r \quad \text{and} \quad a_{j,i} \geq 0 \quad \text{with} \quad \sum_{j=1}^{r} a_{j,i} = 1 \quad \forall i = 1, \ldots, T.$$

In this work we do not focus on the best choice of the Copula parameters. Instead, we simply set $r = 8$ and the coefficients $a_{j,i}$ was generated following a uniform probability distribution with low sparseness. As shown below, this simple choice gives satisfactory results.

5.2.3 Numerical experiments

As already mentioned, it is very expensive computationally to evaluate the probability function, which consists in solving numerically a multidimensional integral. For instance, for evaluating the multivariate normal probability function (A.1) with the `Matlab's` function `mvncdf` one takes almost 40 seconds if $\xi \in \mathbb{R}^{24}$, in the computer described in Section 5.1. In order to compute a subgradient of such a probability function, the function `mvncdf` needs to be called 23 $(T-1)$ times, see Theorem A.3. An alternative to the `mvncdf` function is the routine `mvNcdf` recently developed by Botev [12]. We have verified numerically that Botev's function is around 20 times faster than `mvncdf` to evaluate multivariate normal probability functions. The following results are obtained with the Botev's function.

We solved the power system management problem for $T = 12$ (half day) and $T = 24$ (one day). For dimension $T = 12$, we solved both problems (5.7) and (5.10). For dimension $T = 24$ was not possible to solve (5.7) within one hour CPU time, given the considered computer and softwares. Then we solved (5.10) with $T = 24$ and we checked the probability constraint of (5.7). The results are reported below.

<u>Numerical results for $T = 12$.</u>

The chosen day was Wednesday and solved for $p = 0.8$ and $p = 0.9$ utilizing all solvers.

The first image on Figure 5.3 illustrates the CPU time required by all solvers for both problems. Problem (5.7) (with the multivariate normal probability function) and Problem (5.10) (with copula) with parameter $p = 0.8$ and $p = 0.9$. In this figure, the CPU time spent by all solvers are summed to facilitate visualization. Detailed information about the optimum value and time spent by each method are reported on Table 5.4 and Table 5.5[5]. The CPU time spent to solve problem (5.10) was approximately 113 times faster than utilized to solve problem (5.7).

The number of master subproblems (MILP or MIQP) solved by each method was

[5]We recall that is a maximization problem.

smaller when using copula as shown in the middle image on Figure 5.3. The last image on Figure 5.3 represents the number of oracle calls. This example, shows that it is very expensive to solve numerically the multidimensional integral even for small dimensions.

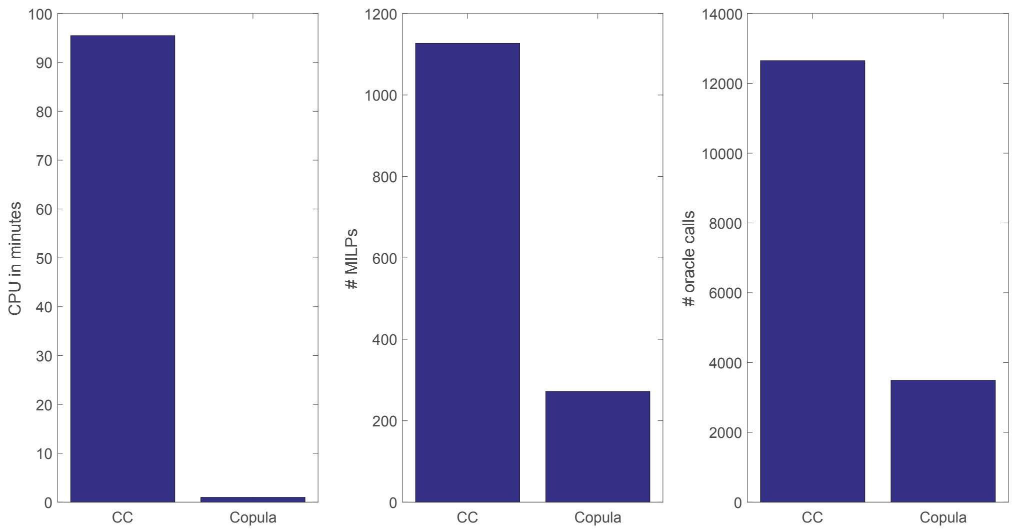

Figure 5.3: Numerical results for problem with $T = 12$ on Wednesday : CPU time, oracle's calls and number of solved mixed-integer subproblems.

Table 5.4: Optimal value and CPU time for OA, $\texttt{OA}_1$ and $\texttt{OA}_\infty$.

	OA			$\texttt{OA}_1$			$\texttt{OA}_\infty$		
	p	optimal value	CPU time	p	optimal value	CPU time	p	optimal value	CPU time
CC	0.8	21234.15	493.69	0.8	21232.02	553.33	0.8	21232.46	680.89
Copula	0.8	21191.96	8.50	0.8	21191.98	3.66	0.8	21191.98	4.44
CC	0.9	21102.19	543.72	0.9	21101.25	431.88	0.9	21099.66	661.35
Copula	0.9	21075.08	7.71	0.9	21075.09	3.48	0.9	21075.07	2.61

Table 5.5: Optimal value and CPU time for $\texttt{OA}_2$, ECPM and ELBM.

	$\texttt{OA}_2$			ECPM			ELBM		
	p	optimal value	CPU time	p	optimal value	CPU time	p	optimal value	CPU time
CC	0.8	21232.21	752.50	0.8	21232.65	451.04	0.8	21233.48	142.82
Copula	0.8	21191.98	4.13	0.8	21192.21	3.77	0.8	21192.07	6.22
CC	0.9	21101.12	401.89	0.9	21101.03	454.38	0.9	21100.49	161.71
Copula	0.9	21075.11	4.20	0.9	21075.79	4.01	0.9	21075.93	6.67

Finally, Table 5.6 demonstrates the quality of the solution. The solution obtained using copulas is the same as the optimal solution with a relative error maximum of 0.2% which is acceptable as tolerance error.

Table 5.6: Quality of solution for T=12.

	p	optimal value	estimate value	relative error
OA	0.8	21234.15	21191.96	0.2%
	0.9	21102.19	21075.08	0.1%
$\texttt{OA}_1$	0.8	21232.02	21191.98	0.2%
	0.9	21101.25	21075.09	0.1%
$\texttt{OA}_\infty$	0.8	21232.46	21191.98	0.2%
	0.9	21099.66	21075.07	0.1%
$\texttt{OA}_2$	0.8	21232.21	21191.98	0.2%
	0.9	21101.12	21075.11	0.1%
ECPM	0.8	21232.65	21192.21	0.2%
	0.9	21101.03	21075.79	0.1%

ELBM	0.8	21232.48	21192.07	0.2%
	0.9	21100.49	21075.93	0.1%

<u>Numerical results for $T = 24$.</u>

In this dimension is not possible to solve problem (5.7) anymore. We solved problem (5.10) by $T = 24$ using Zhangs copula with parameters described above. Figure 5.4 shows the performance profiles [23] of solvers `OA`, $\texttt{OA}_1$, $\texttt{OA}_\infty$, $\texttt{OA}_2$, `ECPM` and `ELBM`.

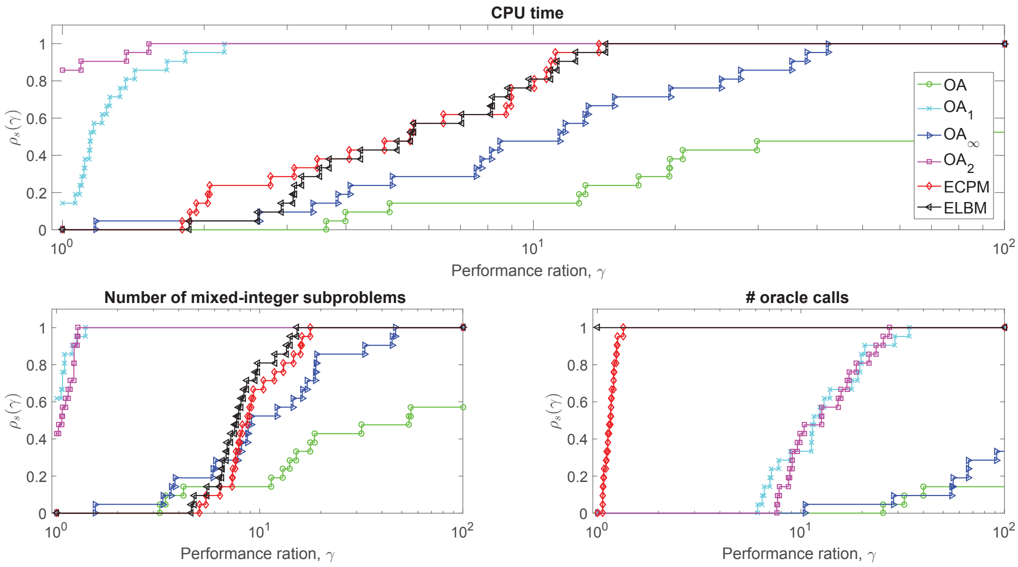

Figure 5.4: Performance profile on 21 problems (logarithmic scale): CPU time, oracle's calls and number of solved mixed-integer subproblems.

The first image on Figure 5.4 corresponds to the performance profiles of CPU time required by the methods on all instances, i.e. problem (5.10) with $p = 0.8, 0.9$ and 0.95. The image shows that the fastest and more robust solver in this application was $\texttt{OA}_2$, followed by $\texttt{OA}_1$. Solvers `ECPM` and `ELBM` had a similar good performance. Solver $\texttt{OA}_\infty$ ranked fifth and the classic `OA` was the worst solver in this application. Notice that $\texttt{OA}_2$ was the fastest solver in around 87.5% of the problems while $\texttt{OA}_1$ was faster in the remaining problems. For 21 problems, all solvers were able to solve problem (5.10) within 1 hour, except `OA`. Solver `OA` failed to solve 9 problems although it almost hit the solution.

The left-bottom image on Figure 5.4 reports the solvers performance considering the number of master subproblems (MILP or MIQP) solved by each method. With respect to this attribute, solver $\texttt{OA}_2$ was the most efficient method even though it require solving MIQP instead of MILP subproblem. It was closely followed by $\texttt{OA}_1$. The methods $\texttt{OA}_\infty$, `ECPM` and `ELBM` had solved approximately 10 times more MILPs subproblems than solver $\texttt{OA}_2$. As expected the `OA` method solved more MILPs in this problem.

The right-bottom image on Figure 5.4 demonstrates the performance profiles considering the number of oracle calls. Regarding this attribute the ELBM and ECPM methods provided better performances than OA solvers corroborating to [16].

Table 5.7 shows the optimal value for all algorithms with $p = 0.8, p = 0.9$ and $p = 0.95$ respectively. For example, if the decision maker wants to supply the local community demand with probability $p = 0.8$ on Tuesday the profit will be $R\$\,43504$. Changing the probability to $p = 0.9$ the profit will be $R\$\,43076$ and finally it will be $R\$\,42767$ for probability $p = 0.95$. These numbers demonstrates that the profit is inversely proportional to the level p of demand satisfaction, i.e. for higher probabilities the lower is the profit. It is important to note that profits are higher in business days because it is directly proportional to demand as well.

Table 5.7: Estimate optimal value for $p \in \{0.8, 0.9, 0.95\}$. The asterisk * stands for unsolved problem (within one hour).

	Day	OA	$\mathtt{OA}_1$	$\mathtt{OA}_\infty$	$\mathtt{OA}_2$	ECPM	ELBM
p=0.8	Tuesday	43504.01	43503.99	43504.01	43504.02	43504.96	43504.33
	Wednesday	43704.87	43704.88	43704.87	43704.88	43705.41	43705.04
	Thursday	43137.90	43137.90	43137.93	43135.39	43135.80	43137.05
	Friday	41619.99*	41690.12	41690.14	41684.85	41690.42	41690.37
	Saturday	41023.31	41023.34	41023.32	41023.33	41023.53	41023.50
	Sunday	34660.05	34660.07	34660.08	34660.04	34660.48	34660.39
	Monday	38443.81*	38489.47	38489.49	38489.47	38489.80	38490.13
p=0.9	Tuesday	43014.20*	43075.82	43075.89	43075.86	43060.45	43077.03
	Wednesday	43368.72	43368.71	43368.72	43368.72	43369.61	43370.00
	Thursday	42784.96	42785.09	42785.10	42779.33	42786.37	42786.15
	Friday	41286.48*	41356.63	41356.66	41356.64	41357.52	41357.16
	Saturday	40857.38	40857.37	40857.34	40857.37	40857.88	40857.88
	Sunday	34466.99	34466.98	34466.99	34466.99	34467.35	34467.15
	Monday	38092.81*	38138.46	38138.53	38138.54	38139.48	38139.22
p=0.95	Tuesday	42706.29*	42767.90	42767.88	42767.89	42769.41	42769.65
	Wednesday	42951.92	42951.68	42951.87	42952.09	42953.43	42955.49
	Thursday	42341.11*	42395.79	42395.64	42392.93	42395.73	42396.77
	Friday	40994.54*	41064.56	41064.56	41064.65	41066.61	41065.52
	Saturday	40690.00	40689.99	40689.95	40689.97	40683.51	40688.40
	Sunday	34303.96	34303.89	34303.94	34303.96	34304.70	34303.63
	Monday	37789.85*	37835.55	37835.62	37835.39	37837.61	37836.54

The obtained decision for the binary variables representing the on/off the turbines are shown on Figure 5.5. Results depend on day and on the volume of water in the reservoir. If the amount of water is abundant then the optimal solution is to turn on the turbine all time independently of the price. However, for the purpose of this work the assumption is that the amount of the water is limited leading to a solution that is no longer trivial.

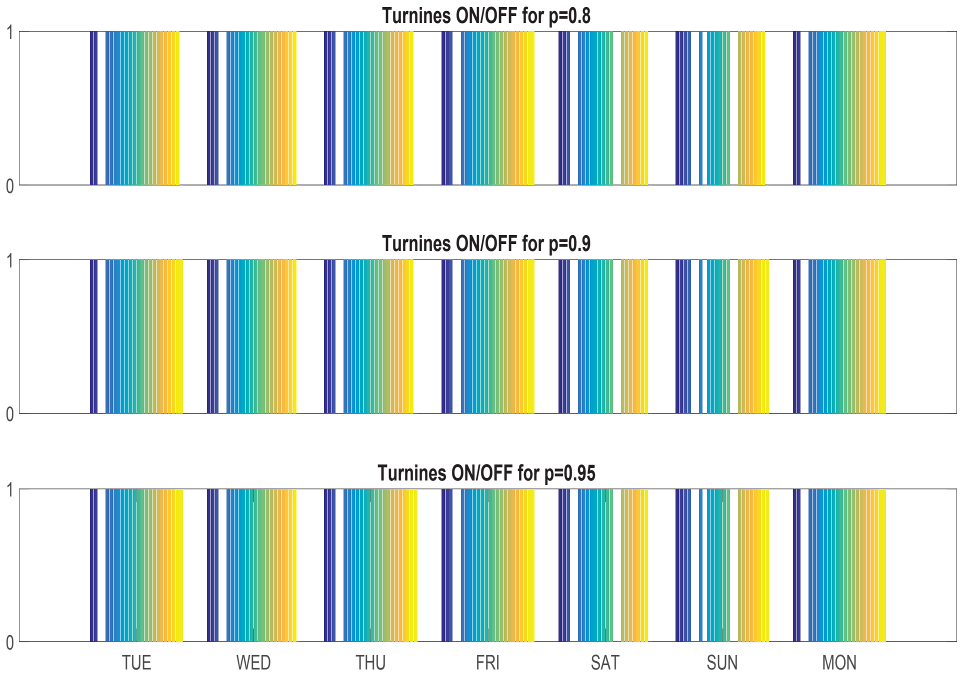

Figure 5.5: Solution ON/OFF turbines

Table 5.8 complements information shown on Figure 5.4 and informs the CPU time spent by every solver and the number of MILPs subproblems solved by each of them. Regularized solvers $\mathtt{OA}_2$ and $\mathtt{OA}_1$ were the best ones on these instances.

Table 5.8: Number of MILPs and CPU time for $p \in \{0.8, 0.9, 0.95\}$

	Day	OA		$\mathtt{OA}_1$		$\mathtt{OA}_\infty$		$\mathtt{OA}_2$		ECPM		ELBM	
		k	CPU	k	CPU	k	CPU	k	CPU	k	CPU	k	CPU
p=0.8	Tuesday	1175	3395.79	39	141.08	539	1485.71	37	113.72	272	462.65	253	623.40
	Wednesday	502	1327.55	44	140.20	147	349.76	47	102.97	281	563.28	242	442.22
	Thursday	967	2958.82	57	174.08	452	1235.68	52	151.84	283	528.34	240	442.62
	Friday	908	3600.89*	41	173.59	781	2971.03	48	152.14	328	733.75	309	848.80
	Saturday	31	36.27	9	9.08	53	45.53	11	12.41	71	16.29	59	16.87
	Sunday	197	407.31	13	21.33	79	79.91	13	19.61	99	36.56	90	68.93
	Monday	653	3605.96*	23	96.57	1073	3347.86	28	87.82	338	787.99	273	618.75
p=0.9	Tuesday	1204	3600.92*	32	179.63	390	1225.37	34	107.71	379	1081.07	305	873.53
	Wednesday	533	1510.74	41	152.42	157	465.13	52	120.90	300	672.95	267	621.72
	Thursday	1031	3425.71	73	213.68	472	1329.76	58	176.13	294	546.93	276	650.54
	Friday	820	3603.04*	42	146.48	793	3203.98	43	128.13	344	825.08	324	1050.28
	Saturday	29	37.92	9	10.44	33	27.35	11	15.92	80	21.23	71	32.49
	Sunday	778	3582.22	15	46.50	461	1239.08	14	35.07	127	67.49	101	91.23
	Monday	646	3609.38*	23	96.59	373	1288.81	28	86.89	365	968.77	312	946.39
p=0.95	Tuesday	1165	3601.44*	32	204.71	394	1310.39	23	112.16	372	1195.34	325	1097.20
	Wednesday	717	2129.51	56	181.47	398	1081.85	51	127.46	389	1144.42	323	1124.83
	Thursday	1032	3600.68*	49	219.76	374	1280.56	41	99.64	382	1086.66	343	1226.53
	Friday	771	3605.22*	42	123.34	789	3045.20	48	110.58	365	968.78	340	1244.15
	Saturday	38	94.58	9	19.13	14	22.51	10	20.93	118	52.96	88	59.09
	Sunday	761	3401.13	15	48.91	123	356.63	14	45.96	146	94.46	121	148.09
	Monday	622	3600.82*	22	98.68	986	3580.10	28	84.72	391	1165.09	336	1208.03
	Sum	14580	15.20 h	686	0.69 h	8881	8.05 h	691	0.53 h	5724	3.62 h	4998	3.73 h

Table 5.7 shows the optimal value for problem (5.10). Next step is to verify if this solution is a feasible point for problem (5.7). In order to have this confirmation it is necessary to check the true constraint at the obtained vector of decisions. The result

should be equal or greater than stipulated value p to ensure that the obtained vector is a feasible decision. Table 5.9 shows the estimated probability (EP) and the true probability (TP), obtained by evaluating the obtained solution with the Botev's code. The table shows that all points found by solving (5.10) are considered feasible points for problem (5.7). These results allow to estimate a good solution for problem (5.7) without using the hard chance constraint.

Table 5.9: Real probability and estimate probability (via copula).

	Day	`OA`		$\mathtt{OA}_1$		$\mathtt{OA}_\infty$		$\mathtt{OA}_2$		`ECPM`		`ELBM`	
		TP	EP	TP	EP	TP	EP	TP	EP	TP	EP	TP	EP
p=0.8	Tuesday	0.83	0.80	0.83	0.80	0.83	0.80	0.83	0.80	0.83	0.80	0.83	0.80
	Wednesday	0.84	0.80	0.84	0.80	0.84	0.80	0.84	0.80	0.84	0.80	0.84	0.80
	Thursday	0.84	0.80	0.84	0.80	0.84	0.80	0.84	0.80	0.84	0.80	0.84	0.80
	Friday	0.84	0.80	0.84	0.80	0.84	0.80	0.84	0.80	0.84	0.80	0.84	0.80
	Saturday	0.85	0.80	0.85	0.80	0.85	0.80	0.85	0.80	0.85	0.80	0.85	0.80
	Sunday	0.83	0.80	0.83	0.80	0.83	0.80	0.83	0.80	0.83	0.80	0.83	0.80
	Monday	0.84	0.80	0.83	0.80	0.84	0.80	0.84	0.80	0.84	0.80	0.83	0.80
p=0.9	Tuesday	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90
	Wednesday	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90
	Thursday	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90
	Friday	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90
	Saturday	0.92	0.90	0.92	0.90	0.92	0.90	0.92	0.90	0.92	0.90	0.92	0.90
	Sunday	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90
	Monday	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90	0.91	0.90
p=0.95	Tuesday	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
	Wednesday	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
	Thursday	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
	Friday	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
	Saturday	0.96	0.95	0.96	0.95	0.96	0.95	0.96	0.95	0.96	0.95	0.96	0.95
	Sunday	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
	Monday	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95

Conclusion

Technology growth and scientific development in the world create a high number of problems that demands mathematical and computational assistance to be solved. Examples of complex problems are energy outlook, oil reserves estimation, the impact of wind in human life (power generation), effects of climate and so on. The characteristics of these problems are that in order to solve it is necessary to apply heavy math and tremendous computational effort. Most of the problems contains uncertainties and integer variables that lead to optimization problems with mixed-integer variables and chance-constraints. This justifies the relevance of the theme considered in this thesis not only to the industrial/productive sector but also to the (applied) mathematical community.

In this work we have studied mixed-integer optimization problems with chance constraints. It is well known in the mathematical community that mixed-integer programming and stochastic programming constitute two very active and challenging research areas. Therefore, the combination of mixed-integer variables and chance-constraints in decision/planning problems lead to optimization programs that are both mathematically and computationally difficult to solve. The setting is even more complicated when differentiability is absent, a common situation in real-life applications.

In order to deal with more general mixed-integer convex programs we have extended the well-known Outer-Approximation algorithm to deal with nonsmooth objective and constraint functions. In order to obtain finite convergence in this more general setting, the linearizations must be computed by using particular subgradients of the nonsmooth functions. To accomplish this task, we have presented a bundle method algorithm that not only solves the resulting OA's nonlinear subproblems but also computes (at no extra-cost) subgradients that satisfy the (nonlinear subproblem's) KKT system, yielding thus convergence of OA algorithms. The bundle algorithm considers an exact penalization function that is employed only to choose stability centers, and as result, the algorithm is not hindered by any large penalization value. This theoretical contribution has been published in the article [20].

We have also presented an outer-approximation algorithmic pattern that employs the given bundle algorithm and that possesses quite some freedom in defining new integer iterates. Such flexibility in choosing trial points opens the way to employ regularization strategies as an attempt to reduce the number of outer-approximation iterations and, as a result, the number of OA's subproblems to be solved.

Concerning the chance-constraint formulation, we have recalled in this work some known results on generalized convexity, and have assessed the computational cost of evaluating probability functions in a practical problem. Furthermore, in order to approximate the probability function, we have investigated a family of (nonsmooth) Copulae and proved some useful generalized convexity properties. Numerical experiments on a fictitious power system management have shown that the copula approach presents itself as an interesting tool for approximating the probabilistic function in a chance-constrained program.

We have numerically assessed the performance of several variants of the proposed OA algorithm on two families of nonsmooth mixed integer chance-constraints problems arising from power management. In the first class of problems the probability distribution is discrete (and finite), and therefore the chance constraint has been modeled with the help of linear functions and binary variables. The second family of problems deals with continuous probability distribution, whose probability values are computed numerically by solving a multidimensional integral. Overall, the obtained numerical results suggest that the proposed regularized variants of the given nonsmooth OA algorithm can provide an effective reduction of the number of outer-approximation iterations required to solve convex mixed-integer programs.

Appendix A

Multivariate normal distribution

In this appendix, a brief review of multivariate normal distribution and its properties are presented. These results were used to solve computationally problem (5.7).

Definition A.1. *A random vector $\xi \in \mathbb{R}^m$ follows a multivariate normal distribution with mean vector μ and covariance matrix Σ if its cumulative distribution function is given by*

$$F_\xi(z_1, \ldots, z_m) = \int \frac{1}{\sqrt{|2\pi\Sigma|}} \exp(-\frac{1}{2}(z-\mu)'\Sigma^{-1}(z-\mu))dw, \tag{A.1}$$

where $z \in \mathbb{R}^m, \mu \in \mathbb{R}^m$ and $\Sigma \in \mathbb{R}^m \times \mathbb{R}^m$. The symbol of integral in (A.1) *means a multiple integral of dimension m. The standard notation is $\xi \sim N(\mu, \Sigma)$.*

The function (A.1) is differentiable if the covariance matrix Σ is positive definite (see Theorem A.3 below).

Theorem A.1. *Suppose that $\xi \in \mathbb{R}^m$ follows a multivariate normal distribution $N(\mu, \Sigma)$. Let be $\delta = A\xi + a$ where $A \in \mathbb{R}^p \times \mathbb{R}^m$ and $a \in \mathbb{R}^m$. Then $\delta \sim N(\bar{\mu}, \bar{\Sigma})$ where*

$$\bar{\mu} = A\mu + a \quad \text{and} \quad \bar{\Sigma} = A\Sigma A^T. \tag{A.2}$$

Proof: See [65, Theorem 3.3.3]. ∎

Suppose that $\xi \sim N(\mu, \Sigma)$. As $-\xi = -I\xi$, by Theorem A.1 we have that $-\xi \sim N(-\mu, \Sigma)$.

Theorem A.2. *Suppose that $\xi \in \mathbb{R}^m$ follows a multivariate normal distribution $N(\mu, \Sigma)$. Let be*

$$\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \quad , \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad \text{and} \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

where $\xi_1 \in \mathbb{R}^q, \xi_2 \in \mathbb{R}^{m-q}, \mu_1 \in \mathbb{R}^q, \mu_2 \in \mathbb{R}^{m-q}, \Sigma_{11} \in \mathbb{R}^q \times \mathbb{R}^q, \Sigma_{12} \in \mathbb{R}^q \times \mathbb{R}^{m-q}, \Sigma_{21} \in$

$\mathbb{R}^{m-q} \times \mathbb{R}^q$ and $\Sigma_{22} \in \mathbb{R}^{m-q} \times \mathbb{R}^m$ with $m > q \geq 1$. Then

$$\xi_1 \sim N(\mu_1, \Sigma_{11}) \quad \textit{and} \quad \xi_2 \sim N(\mu_2, \Sigma_{22}). \tag{A.3}$$

Proof: See [65, Theorem 3.3.1]. ■

Suppose that diagonal of matrix Σ is the vector $[\sigma_1^2, \sigma_2^2, \ldots, \sigma_m^2]$. Then, by Theorem A.2 we have $-\xi_i \sim N(-\mu_i, \sigma_i^2)$. In other words, the unidimensional margins of a multivariate normal distribution is also normal.

Theorem A.3. *Suppose that $\xi \in \mathbb{R}^m$ follows a multivariate normal distribution with mean vector $\mu \in \mathbb{R}^m$ and positive covariance matrix $\Sigma \in \mathbb{R}^m \times \mathbb{R}^m$. Then the distribution function $F_\xi(z) = P[\xi \leq z]$ is continuously differentiable and in any fixed $z \in \mathbb{R}^m$ the following holds for arbitrary $i = 1, \ldots, m$:*

$$\frac{\partial F_\xi}{\partial z_i}(z) = f_{\xi_i}(z_i) F_{\tilde{\xi}(z_i)}(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_m), \tag{A.4}$$

where f_{ξ_i} is the one dimensional normal density associated to random variable ξ_i given by

$$f_{\xi_i}(z_i) = \frac{1}{\sqrt{2\pi}\sigma_{ii}} e^{\frac{1}{\sigma_{ii}^2}(z_i - \mu_i)^2}$$

and $F_{\tilde{\xi}(z_i)}$ is the cumulative distribution function associated to random vector $\tilde{\xi}(z_i)$ with mean vector $\hat{\mu} \in \mathbb{R}^{m-1}$ and covariance matrix $\hat{\Sigma} \in \mathbb{R}^{m-1} \times \mathbb{R}^{m-1}$. Let D_m^i denote the $(m-1) \times m$ matrix obtained from the $m \times m$ identity matrix deleting the ith row. Then

$$\hat{\mu} = D_m^i(\mu + \Sigma_{ii}^{-1}(z_i - \mu_i)\Sigma_i) \quad \textit{and} \quad \hat{\Sigma} = D_m^i(\Sigma - \Sigma_{ii}^{-1}\Sigma_i\Sigma_i^T)(D_m^i)^T, \tag{A.5}$$

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

Proof: See [73, Theorem 1]. ■

Theorem A.3 gives the formulas to gradient of function F_ξ associate to measure P of problem (5.7). Note if the dimension of ξ is m then it is necessary to evaluate m multiple integrals of dimension $m - 1$ (one multiple integral for each coordinate of gradient).

References

[1] Ambrosio, C., Frangioni, A., Liberti, L., and Lodi, A. On interval-subgradient and no-good cuts. *Operations Research Letters 38*, 5 (2010), 341 – 345.

[2] Amor, H. M. T. B., Desrosiers, J., and Frangioni, A. On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics 157*, 6 (2009), 1167 – 1184. Reformulation Techniques and Mathematical Programming.

[3] Arnold, T., Henrion, R., Möller, A., and Vigerske, S. *A mixed-integer stochastic nonlinear optimization problem with joint probabilistic constraints.* Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Institut für Mathematik, 2013.

[4] Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., and A., M. Mixed-integer nonlinear optimization. *Acta Numerica 22* (5 2013), 1–131.

[5] Benders, J. F. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik 4*, 1 (Dec 1962), 238–252.

[6] Bertsimas, D., Brown, D., and Caramanis, C. Theory and applications of robust optimization. *SIAM Review 53*, 3 (2011), 464–501.

[7] Bienstock, D., and Mattia, S. Using mixed-integer programming to solve power grid blackout problems. *Discrete Optimization 4*, 1 (2007), 115 – 141. Mixed Integer Programming.

[8] Birge, J., and Louveaux, F. *Introduction to Stochastic Programming.* Springer-Verlag, New York, 2011.

[9] Bonami, P., Biegler, L., Conn, A., Cornuéjols, G., Grossmann, I., Laird, C., Lee, J., Lodi, A., Margot, F., Sawaya, N., and Wächter, A. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization 5*, 2 (2008), 186 – 204. In Memory of George B. Dantzig.

[10] Bonnans, J., Gilbert, J., Lemaréchal, C., and Sagastizábal, C. *Numerical Optimization. Theoretical and Practical Aspects.* Universitext. Springer-Verlag Berlin Heidelberg, Berlin, 2006. Second edition, xiv+490 pp.

[11] Bonvin, G., Demassey, S., Le Pape, C., Maïzi, N., Mazauric, V., and Samperio, A. A convex mathematical program for pump scheduling in a class of branched water networks. *Applied Energy 185* (2017), 1702 – 1711. Clean, Efficient and Affordable Energy for a Sustainable Future.

[12] Botev, Z. I. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society Series B 79*, 1 (2017), 125–148.

[13] Charnes, A., and Cooper, W. W. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research 11*, 1 (1963), 18–39.

[14] Charnes, A., Cooper, W. W., and Symonds, G. H. Cost horizons and certainty equivalents: An approach to stochastic programming of heating oil. *Management Science 4*, 3 (1958), 235–263.

[15] Choroś, B., Ibragimov, R., and Permiakova, E. Copula estimation. In *Copula Theory and Its Applications* (2010), Springer Berlin Heidelberg, pp. 77–91.

[16] de Oliveira, W. Regularized optimization methods for convex minlp problems. *TOP 24*, 3 (Oct 2016), 665–692.

[17] de Oliveira, W., and Sagastizábal, C. Level bundle methods for oracles with on-demand accuracy. *Optimization Methods and Software 29*, 6 (2014), 1180–1209.

[18] de Oliveira, W., Sagastizábal, C., and Lemaréchal, C. Convex proximal bundle methods in depth: a unified analysis for inexact oracles. *Mathematical Programming 148*, 1 (Dec 2014), 241–277.

[19] de Oliveira, W., and Solodov, M. A doubly stabilized bundle method for nonsmooth convex optimization. *Mathematical Programming 156*, 1 (Mar 2016), 125–159.

[20] Delfino, A., and de Oliveira, W. Outer-approximation algorithms for nonsmooth convex minlp problems. *Optimization 67*, 6 (2018), 797–819.

[21] Dentcheva, D., and Martinez, G. Regularization methods for optimization problems with probabilistic constraints. *Mathematical Programming 138*, 1 (Apr 2013), 223–251.

[22] Dentcheva, D., Prékopa, A., and Ruszczynski, A. Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming 89*, 1 (Nov 2000), 55–77.

[23] Dolan, E. D., and Moré, J. J. Benchmarking optimization software with performance profiles. *Mathematical Programming 91*, 2 (Jan 2002), 201–213.

[24] Duran, M. A., and Grossmann, I. E. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming 36*, 3 (Oct 1986), 307–339.

[25] Durante, F., and Sempi, C. *Principe of Copula Theory*. Taylor & Francis, 2015.

[26] Eronen, V., Mäkelä, M. M., and Westerlund, T. On the generalization of ecp and oa methods to nonsmooth convex minlp problems. *Optimization 63*, 7 (2014), 1057–1073.

[27] Fábián, C. Bundle-type methods for inexact data. *Central European Journal of Operations Research 8* (2000), 35–55.

[28] Fletcher, R., and Leyffer, S. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming 66*, 1 (Aug 1994), 327–349.

[29] Frangioni, A., and Gentile, C. Perspective cuts for a class of convex 0–1 mixed integer programs. *Mathematical Programming 106*, 2 (Apr 2006), 225–236.

[30] Geoffrion, A. M. Generalized benders decomposition. *Journal of Optimization Theory and Applications 10*, 4 (Oct 1972), 237–260.

[31] Goldberg, N., Leyffer, S., and Safro, I. Optimal response to epidemics and cyber attacks in networks. *Networks 66*, 2 (2015), 145–158.

[32] Grossmann, I. E. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering 3*, 3 (Sep 2002), 227–252.

[33] Gupta, O. K., and Ravindran, A. Branch and bound experiments in convex nonlinear integer programming. *Management Science 31*, 12 (1985), 1533–1546.

[34] Gurobi Optimization, I. Gurobi optimizer reference manual, 2016.

[35] Hemmecke, R., Köppe, M., Lee, J., and Weismantel, R. Nonlinear integer programming. In *50 Years of Integer Programming 1958-2008*, M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, Eds. Springer Berlin Heidelberg, 2010, pp. 561–618.

[36] Henrion, R., and Möller, A. Optimization of a continuous distillation process under random inflow rate. *Computers & Mathematics with Applications 45*, 1 (2003), 247 – 262.

[37] Henrion, R., and Strugarek, C. Convexity of chance constraints with independent random variables. *Computational Optimization and Applications 41*, 2 (Nov 2008), 263–276.

[38] Henrion, R., and Strugarek, C. *Convexity of chance constraints with dependent random variables: the use of copulae*. Springer, 2011.

[39] Hiriart-Urruty, J. B., and Lemaréchal, C. *Convex Analysis and Minimization Algorithms*. No. 305-306 in Grundlehren der mathematischen Wissenschaften. Springer-Verlag Berlin Heidelberg, 1993. (two volumes).

[40] Hiriart-Urruty, J. B., and Lemaréchal, C. *Convex Analysis and Minimization Algorithms II*, 2nd ed. No. 306 in Grundlehren der mathematischen Wissenschaften. Springer-Verlag, 1996.

[41] Karuppiah, R., and Grossmann, I. E. Global optimization for the synthesis of integrated water systems in chemical processes. *Computers & Chemical Engineering 30*, 4 (2006), 650 – 673.

[42] Kelley, J. J. E. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics 8*, 4 (1960), 703–712.

[43] Kiwiel, K. C. Exact penalty functions in proximal bundle methods for constrained convex nondifferentiable minimization. *Mathematical Programming 52*, 1 (May 1991), 285–302.

[44] Kiwiel, K. C., and Lemaréchal, C. An inexact bundle variant suited to column generation. *Mathematical Programming 118*, 1 (Apr 2009), 177–206.

[45] Lemaréchal, C., Nemirovskii, A., and Nesterov, Y. New variants of bundle methods. *Mathematical Programming 69*, 1 (Jul 1995), 111–147.

[46] Leyffer, S. Integrating sqp and branch-and-bound for mixed integer nonlinear programming. *Computational Optimization and Applications 18*, 3 (Mar 2001), 295–309.

[47] Luedtke, J. A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Mathematical Programming 146*, 1 (Aug 2014), 219–244.

[48] Luedtke, J., and Ahmed, S. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization 19*, 2 (2008), 674–699.

[49] Miller, B. L., and Wagner, H. M. Chance constrained programming with joint constraints. *Operations Research 13*, 6 (1965), 930–945.

[50] Morgan, D. R., Eheart, J. W., and Valocchi, A. J. Aquifer remediation design under uncertainty using a new chance constrained programming technique. *Water Resources Research 29*, 3 (1993), 551–561.

[51] Munari, P., and Gondzio, J. Using the primal-dual interior point algorithm within the branch-price-and-cut method. *Computers & Operations Research 40*, 8 (2013), 2026 – 2036.

[52] Nelsen, R. B. *An Introduction to Copulas.* Springer, New York, 2006.

[53] Prékopa, A. On probabilistic constrained programming. *Mathematical Programming Study 28* (1970), 113–138.

[54] Prékopa, A. Contributions to the theory of stochastic programming. *Mathematical Programming 4*, 1 (Dec 1973), 202–221.

[55] Prékopa, A. *Stochastic programming.* Dordrechet, 1995.

[56] Quesada, I., and Grossmann, I. E. An lp/nlp based branch and bound algorithm for convex minlp optimization problems. *Computers & Chemical Engineering 16*, 10 (1992), 937 – 947. An International Journal of Computer Applications in Chemical Engineering.

[57] Rockafellar, R. T. *Convex Analysis*, 1st ed. Princeton University Press, 1970.

[58] Rockafellar, R. T., and Wets, R. J. B. *Variational Analysis.* Springer Verlag Berlin, 2009.

[59] Ruszczyński, A. *Nonlinear Optimization.* Princeton. Princeton University Press, 2006.

[60] Sagastizábal, C. Divide to conquer: decomposition methods for energy optimization. *Mathematical Programming 134*, 1 (Aug 2012), 187–222.

[61] Shapiro, A. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research 209*, 1 (2011), 63 – 72.

[62] Shapiro, A., Dentcheva, D., and Ruszczyński, A. *Lectures on stochastic programming.* MPS-SIAM, 2009.

[63] Sklar, A. Fountions de rapartition á dimentisions et leurs marges. *Publications and Institut de Statistique de Paris 8* (1959), 229–231.

[64] Song, Y., Luedtke, J., and Küçükyavuz, S. Chance-constrained binary packing problems. *INFORMS Journal on Computing 26*, 4 (2014), 735–747.

[65] Tong, Y. L. *The Multivariate Normal Distribution.* Springer Series in Statistic, 1990.

[66] van Ackooij, W. Decomposition approaches for block-structured chance-constrained programs with application to hydro-thermal unit commitment. *Mathematical Methods of Operations Research 80*, 3 (Dec 2014), 227–253.

[67] van Ackooij, W., Berge, V., de Oliveira, W., and Sagastizábal, C. Probabilistic optimization via approximate p-efficient points and bundle methods. *Computers & Operations Research 77* (2017), 177 – 193.

[68] van Ackooij, W., and de Oliveira, W. Level bundle methods for constrained convex optimization with various oracles. *Computational Optimization and Applications 57*, 3 (Apr 2014), 555–597.

[69] van Ackooij, W., Frangioni, A., and de Oliveira, W. Inexact stabilized benders' decomposition approaches with application to chance-constrained problems with finite support. *Computational Optimization and Applications 65*, 3 (Dec 2016), 637–669.

[70] van Ackooij, W., Henrion, R., Möller, A., and Zorgati, R. On probabilistic constraints induced by rectangular sets and multivariate normal distributions. *Mathematical Methods of Operations Research 71*, 3 (Jun 2010), 535–549.

[71] van Ackooij, W., Henrion, R., Möller, A., and Zorgati, R. Joint chance constrained programming for hydro reservoir management. *Optimization and Engineering 15*, 2 (Jun 2014), 509–531.

[72] van Ackooij, W., and Malick, J. Eventual convexity of probability constraints with elliptical distributions. *Mathematical Programming* (Jan 2018).

[73] van Ackooij, W., and Minoux, M. A characterization of the subdifferential of singular gaussian distribution functions. *Set-Valued and Variational Analysis 23*, 3 (Sep 2015), 465–483.

[74] Vielma, J., Ahmed, S., and Nemhauser, G. Mixed integer linear programming formulations for probabilistic constraints. *Operations Research Letters 40*, 3 (2012), 153 – 158.

[75] Wei, Z. Ali, M. M. Convex mixed integer nonlinear programming problems and an outer approximation algorithm. *Journal of Global Optimization 63*, 2 (Oct 2015), 213–227.

[76] Westerlund, T., and Pettersson, F. An extended cutting plane method for solving convex minlp problems. *Computers & Chemical Engineering 19* (1995), 131 – 136. European Symposium on Computer Aided Process Engineering 3-5.

[77] Westerlund, T., and Pörn, R. Solving pseudo-convex mixed integer optimization problems by cutting plane techniques. *Optimization and Engineering 3*, 3 (Sep 2002), 253–280.

[78] Zaourar, S., and Malick, J. Quadratic stabilization of Benders decomposition. working paper or preprint, 2014.

[79] Zhang, Z. On approximation max-stable process and constructing extremal copula functions. *Statistical Inference for Stochastic Process 12* (2009), 89–114.