

Static risk-averse models with applications to mining

Ph.D. student: Gianpiero Canessa Gisseleire
Advisor: Bernardo K. Pagnoncelli

October 12, 2021

Contents

List of Figures	4
List of Tables	5
1 Introduction	6
1.1 Motivation	6
1.2 Acknowledgements	8
2 Bibliographic Review	10
2.1 Binary IIS Algorithm	10
2.2 Risk-averse Ultimate Pit Problem	11
3 Binary IIS Algorithm	13
3.1 Introduction	13
3.2 Infeasible irreducible subsystems	15
3.2.1 The deterministic equivalent formulation	15
3.2.2 Irreducibly infeasible subsystems	16
3.2.3 IIS and CCP Problems	17
3.3 Strategies to solve binary CCP	19
3.3.1 Efficient Upper Bound Generation	19
3.3.2 The choice of ϵ for binary problems	20
3.4 The algorithm	22
3.4.1 Preliminaries	22
3.4.2 IIS branch-and-cut algorithm	23
3.5 Numerical experiments	25
3.5.1 Stochastic Set Covering Problem	25
3.5.2 Vaccine Allocation Problem	28
3.6 Conclusions	31
4 Risk-averse UP problem	33
4.1 Introduction	33
4.2 Ultimate Pit problem and nested pits	34
4.2.1 UP problem	34
4.2.2 Nested pits	36

<i>CONTENTS</i>	3
4.3 Risk measurement	37
4.3.1 Defining risk under ore grade uncertainty	37
4.3.2 The Entropic risk measure	39
4.4 Computational results	39
4.4.1 Small mine	40
4.4.2 Andina case	43
4.5 Conclusions	46
5 Bibliography	47
A Risk-averse UP Problem	51
A.1 Mathematical proofs	51
A.1.1 Risk-nestedness for the UP problem of the entropic measure	51
A.1.2 Proof of Lemma 1	52
A.1.3 Proof of Lemma 2	52
A.2 Other Lemmas and Propositions of interest	53
A.3 Entropic risk objective function linearization	54

List of Figures

1.1	Number of publications in Scopus under the tag <i>stochastic optimization</i> from 2000 to 2019.	6
3.1	The set Q	17
3.2	Performance comparison between instances.	29
4.1	Example of nested pits.	36
4.2	Three blocks.	38
4.3	Topology of the small mine: blue means the block is part of set of precedent blocks of block b	40
4.4	Configuration of the blocks in the small mine.	41
4.5	Configuration of the pits obtained with different values of α and changing the variance-covariance configuration.	42
4.6	Configuration of the pits obtained with different values of β	42
4.7	Negative profits per scenario for the Andina case.	45
4.8	Comparing pits of both models.	45
A.1	Subgradient linear approximation of $\exp(x)$ using $K = 4$ support points where $U = \{-2, -1, 0, 1\}$	55

List of Tables

3.1	Comparison of performance between Gurobi and IISBAC using different sample sizes in scp41	27
3.2	Comparison of performance between Gurobi and IISBAC using different sample sizes in scpa1	28
3.3	Problem sizes for vaccination test instances. $\alpha = 0.05$	29
3.4	Larger samples (1,000 and 2,000) results using $\alpha = 0.05$. * final gap 1.5%.	30
4.1	Comparison of mining preference under CVaR.	38
4.2	Table of descriptive statistics for NP of the out of sample results.	44
4.3	Table of descriptive statistics for Ent of the out of sample results.	44

Chapter 1

Introduction

1.1 Motivation

The majority of real-life optimization problems often have to deal with uncertainty from one or more sources when taking decisions. The simplest method to cope with uncertainty is to transform the random elements into a single number (i.e. commonly the mean value) and solve the problem. However, this simplification does not capture the full characteristic of the uncertain elements, e.g. *deviations*, *tail effects*, and how they can affect our decisions. Stochastic optimization aims to avoid this oversimplification by incorporating uncertainty and the preferences of the decision maker (*risk aversion*) into the optimization modeling and solving process. We can see how this topic is steadily growing in terms of publications in the last 20 years as shown in Figure 1.1: as of July of 2019 a total of 2,228 articles have been published through the year according to Scopus, which is already 53.5% of 2018's total.

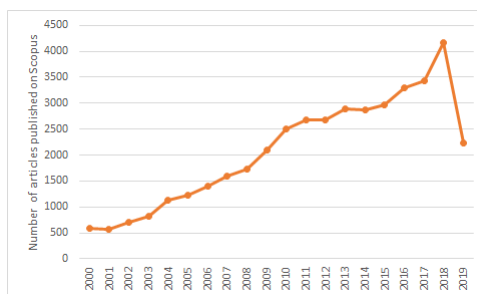


Figure 1.1: Number of publications in Scopus under the tag *stochastic optimization* from 2000 to 2019.

This thesis is centered around the development of methods and algorithms to solve different types of stochastic optimization problems that deal with risk. The first work is on chance-constrained problems (CCP), and the second on risk-averse two-stage stochastic problems (TSSP). The main challenge of stochastic

optimization is that one often generates models that cannot be solved directly: we need to transform it to a tractable deterministic equivalent problem (DEP) and solve it using any of the commercial solvers available. Naïve reformulations into DEPs can, and often will, result in complex and/or large DEPs that current solvers may not be able to solve in an adequate amount of time, or even load in memory due to its size. As an example, the use of optimization in mining—one of the most important economical activities of Chile—is primarily focused in obtaining the exploitation scheduling of a mine, which can often lead to formulations with millions of constraints and decision variables. The large size of these models is one of the reasons why stochastic optimization has experienced a slow adoption in this research community.

These issues have inspired many researchers to find answers to the following questions: are there other ways to formulate and/or solve the same problem? Can we simplify the formulation in order to solve it within a time limit? Can we leverage on the structure of the formulation to generate efficient algorithms to solve the problem? As stated in [24], the continuous growth of computational power has been able to tackle part of these issues, but there still much ground to cover if we wish to rely on technology alone. This is the main motivation of this thesis: generate efficient algorithms and methodologies to cope with risk-averse stochastic programming problems.

In our first paper, we build on the work of [43]. We considered linear binary CCP, an important class of problems which has not received the deserved attention in the literature. We started by formulating a DEP, which relies on auxiliary binary variables. Our main contribution was to use infeasible irreducible subsystems (IIS) for linear binary CCP. We leveraged on modern solver’s capabilities of finding approximate IIS for binary problems, and instead of attacking the DEP directly, our algorithm generated cuts that were added to a master problem at each iteration. Our methodology exploited the fact that we are dealing with binary variables, and we were focused on feasibility and optimality.

Our first set of experiments focused on several instances of a vaccine problem, which has a single chance constraint. We were able to reduce the number of nodes explored in a branch-and-cut tree created by the solver substantially; for large instances the reduction was more than 99%. Furthermore, while resolution time was not our main goal, we solved these large problems faster than a naïve approach using a commercial solver, which in some cases did not close the optimality gap after 24 hours.

Our methods inspired an heuristic to cope with the more challenging case of a joint chance constraint. We considered a probabilistic set covering problem described in [7], and obtain good quality approximations for the optimal value for most instances.

In the second paper of this thesis, we proposed a new formulation to the stochastic version of the ultimate pit (UP) problem. After a stochastic model for the ore grade is inferred through drill samples and techniques such as *kriging*, the mine is then divided into a set of blocks that are a mathematical representation of the deposit. The block model allows one to impose precedence constraints to avoid cave-ins or collapses of the walls of the mine. The UP

problem determines the contour of the mine assuming an infinite availability of resources. The optimal solution of this problem is often used as an input to other approaches that provide a schedule of extraction over time.

The UP problem is commonly solved using the *mean* ore grade of each block, transforming it to a deterministic problem. As discussed before, the use of a single value for each block's ore grade can lead to negative consequences. We formulate the UP problem as a risk-averse TSSP, using the Entropic risk measure (Ent) to represent risk-aversion. Ent is a non-coherent convex risk measure and it is not commonly used in this type of problems. We contrast the use of this risk measure with the more popular Conditional Value-at-Risk (CVaR), which proved to lack desirable properties in the mining context.

We first show results for a self-constructed instance and show how uncertainty can change the final configuration of the solution depending of the risk aversion of the decision maker. We also apply our methodology to Andina, a real world mine of Chile, to show its potential in real life instances. Finally, our conclusions detail how Ent proves to be a adequate tool to capture uncertainty in the stochastic UP problem, how variance and covariance in the ore grade stochastic model can change the problem's solutions, and how changing the risk aversion level is a superior methodology to generate ultimate pits that are used as input in the long-term scheduling heuristics.

1.2 Acknowledgements

It is unfortunate that every single person that was part of this work may not be listed below, as I have many thanks to give to many people responsible for this thesis to exist. However, I can not avoid to pay my respects to those that played a key role in this endeavor:

To Dr. Javiera Barrera, who believed in me almost 7 years ago and gave me the chance of a lifetime.

To Dr. Eduardo Moreno, who proved to be an incredible mentor and guide.

To Dr. Daniel Espinoza, he played a key role in our first paper.

To Dr. Susana Mondschein, who gave me the opportunity to work alongside her in healthcare, which has become an area of interest to me.

To Dr. Lewis Ntaimo, who became a great friend and mentor.

To my friends and classmates: Orlando Rivera, Dr. Sebastián Arpón, Cristián Ramírez, Pablo Ábalos, Felipe Walker and Antonio Lizama, who helped me along the way.

To my parents Ezio Canessa and Jessica Gisseleire, uncle Pablo Gisseleire and aunt Sandra Ansaldo, grandparents Pablo Gisseleire and Cristina Quezada, my cousins Pablo and Francesca and my brothers Enzo and Mauro, that patiently helped me in my hours of need.

To Dr. Leopoldo Bertossi and my aunt Jennifer Gisseleire, who believed this was my true calling ever since I joined college.

To Dr. Roberto Cominetti and Dr. Tito Homem-de-Mello, directors of my Ph. D. program, who helped me in every step of the way, their doors were never closed when I needed their assistance.

To the Anillo Grant ACT 1407 and PienSA Minería 2018 Grant, who believed in our work and passion.

Last but never least:

To Dr. Bernardo Pagnoncelli, who had the patience and disposition to help me along the way and who I consider one of the few reasons I was able to pull this through. A great mentor and friend.

To Mara and T'Challa, my ever present pets who always knew how to make me laugh and relax.

To Natalie, whose love and constant support made this whole endeavor possible to achieve.

In the loving memory of T'Challa, this work is dedicated to his ever loyal companionship.

Chapter 2

Bibliographic Review

2.1 Binary IIS Algorithm

- [44] is the cornerstone of the first part of our research. They proposed a branch-and-cut algorithm for chance-constrained problems (CCP) with continuous variables. By working on the deterministic equivalent formulation of the problem, which can be cast as a mixed integer problem, they were able to reduce the computational effort and time to solve it. Using their work, we set ground for ours by first reproducing their results, and then extending their methodology for CCP with binary variables.
- [41] is the main source of information on general properties of chance-constrained problems, setting the base on our work. Chapter 4 was studied to understand what a chance constrained problem is, and chapter 5 (specifically §5.1) was used as a reference on the use of the sample average approximation (SAA) for CCP.
- [8] was also a source of information on chance-constrained problems approximations using SAA. This work explains in detail how to tune parameters in order to obtain lower and upper bounds on the problem, in a very general setting.
- [13] thoroughly describes and explains what an IIS is and their implications in optimization problems. The author describes different algorithms to find IIS for different classes of problems, such linear, nonlinear, etc.
- [20] propose a simple method to obtain IIS using linear programming, that works only for the continuous case. Their approach finds a polytope such that the support of the vertices are in one-to-one correspondence with the IIS of a given system of inequalities. Tanner and Ntamo make extensive use of this result to obtain the cuts in their paper.
- [32] provide sample size estimates for SAA in chance-constrained problems, guaranteeing that the optimal value and optimal solution of the

approximate problems are close to their deterministic counterparts with high probability. The sample size do not grow significantly if the reliability is increased, and the authors show how the method can be applied to a probabilistic set covering problem, and to a transportation problem. We are using the probabilistic set covering problem to test our methodology.

- [3] and [42] are works on binary chance-constrained problems which also propose cutting techniques for specific types of problems. We consulted these papers in order compare their computational results with our technique. [14]

2.2 Risk-averse Ultimate Pit Problem

- [35] propose a risk measure that is consistent with second degree stochastic dominance. We studied this paper as a first approach to risk-averse measures theory.
- [39] formulate a new approach on optimizing a portfolio of financial instrument to reduce risk. They were able to find a deterministic equivalent formulation to calculate the CVaR as optimization problem. This work is crucial to our understanding on optimization problems using risk measures.
- [14] consider what should the appropriate risk measure for a type of transportation problem. They show that the only measure that satisfies a natural property for this problem is the entropic risk measure. This work inspired us to consider the questions formulated in §4.1.
- [25] propose multistage risk-averse models and discuss the concept of consistency. They show that the so-called expected conditional risk measures are a promising alternative.
- [15] is a empirical investigation of decomposition algorithm to solve stochastic linear programs using risk measures. We studied this as it could prove useful in implementation of decomposition algorithm .
- [19] is one of the base papers studied for our work. The ultimate pit problem is formulated using ore grade uncertainty and results are compared using three different risk measures: Expected value, Conditional Value-at-Risk and ϵ -Modulated Convex-Hull. We used it to understand the ultimate pit problem and to determine its stochastic equivalent for other risk measures not considered in the paper.
- [28]. In this work, instead of ore grade uncertainty, the authors focus on price uncertainty. They propose a robust optimization (RO) model and show that the results are not significantly different to the deterministic case. Possible RO is not the best technique to deal with price uncertainty, and we plan to explore alternatives in the last year of the thesis.

- [34] is another base paper used in this study. They formulate a two-stage stochastic program to solve an open pit mine planning problem, using ore grade uncertainty, capacity constraints and the expected value of net present values. Also, they compare their results using different approaches: obtaining their planning policies using different grades of uncertainty and mining method and comparing them against the expected value with perfect information. This work is our main reference for the stochastic scheduling problem.
- [4] is one of the few papers available where they deal with mining problems using price uncertainty by using two different risk measures. The authors focus on underground operations, and use multistage stochastic programming to solve the problem. They consider risk aversion in the objective function, using the value-at-risk and variations of the conditional value-at-risk as risk measures to the nested formulation of risk averse models, and apply their findings to a pension fund problem.
- [22] advocates for the use of chance constraints in mining scheduling problems. The authors consider ore uncertainty and the chance-constraint is imposed to enforce a minimum and maximum amount of ore production at each period with high probability. The authors propose a formulation but do not solve, so one cannot know what are the changes in terms of policy when using chance constraints.
- [33] study the ultimate pit mining problem and propose a methodology to estimate the profit obtained by each block using data from drills. In their approach, profit is computed for each scenario, and then the average profit is calculated. Results show that such methodology gives superior returns when compared to the traditional case of using the average grade estimation on the profit function. This paper will serve as a basis for our work, and we want to expand their findings by including uncertainty into the optimization problem, and using risk measures to understand how the ultimate pit changes.

Chapter 3

An algorithm for binary chance-constrained problems using IIS

3.1 Introduction

Chance-constrained programming (CCP) is a modelling framework to address optimization problems under uncertainty. The resulting problems are notoriously difficult to solve, mainly due to the lack of convexity in the general case. Even for continuous problems, explicitly evaluating whether a candidate point is feasible is challenging, and Monte-Carlo methods are often employed. Aside from very special cases, it is challenging to solve CCP to optimality, and approximations need to be considered.

CCP was first proposed in [12] and has been extensively studied since then. We believe the main reason for its popularity is that CCP represents a very natural and intuitive way for modelling uncertainty. In the typical case, the decision maker optimizes a deterministic function, and randomness is present in the constraints, with known distribution. A given point is feasible whenever the constraints are satisfied with (at least) some pre-defined probability level. In other words, the search of the optimal solution is restricted to points that satisfy the random constraints for a “large” percentage of the realizations of the random vector.

Aside from being an intuitive modelling tool, CCP is simpler than two-stage stochastic programming problems in terms of parameter specification. In some situations there is no recourse action after a decision is made, and even when recourse is available, estimating the second stage cost coefficients can be challenging. Finally, CCP captures risk aversion since a chance constraint can be viewed as a Value-at-Risk (VaR) constraint.

A general linear CCP problem can be written as follows:

$$\begin{aligned}
& \min f(x) \\
& \text{s.t. } Ax \geq b, \\
& \quad P\{T(\omega)x \geq h(\omega)\} \geq 1 - \alpha, \\
& \quad x \in \mathcal{X}.
\end{aligned} \tag{3.1.1}$$

The objective function $f(\cdot)$ is convex in $x \in \mathbb{R}^n$, and in most applications it is linear. The constraints defined by matrix $A_{m_1 \times n}$ and vector $b_{m_1 \times 1}$ represent the m_1 deterministic constraints of the problem. For instance in a portfolio problem one could impose no short sales, and that the sum of all investments has to be equal to one. We assume ξ is a d -dimensional random vector with probability distribution P supported on a set $\Xi \subset \mathbb{R}^d$, T is a $m_2 \times n$ matrix and h is a $m_2 \times 1$ vector. When $m_2 > 1$ we have joint chance constraints, which will be dealt with separately in this paper since this case needs special treatment. The chance constraint has to be satisfied with probability at least $1 - \alpha$, where $\alpha \in [0, 1)$ is the desired reliability level defined by the decision maker. Finally, the set $\mathcal{X}$ can be continuous, integer or binary.

Several algorithms have been proposed for different versions of the problem. When matrix T has dimension $1 \times n$ and ξ follows a multivariate normal distribution, the chance constraint can be converted into a second order conic constraint, which can be solved efficiently by off-the-shelf solvers ([1], [27], [45]). For problems where randomness is only present on vector h , the concept of p -efficient points can be applied, which allows the construction of tractable equivalent formulations for the CCP (see [17] for the discrete case and [38] for the continuous one). The method was extended by [29] to include mixed-integer variables.

The Sample Average Approximation (SAA) is a popular approach whose advantages are that both matrix T and vector h can be random, and that the distribution is arbitrary, as long as samples can be obtained from it ([32], [36]). When the original distribution is continuous, or discrete with a very large number of scenarios, SAA consists in generating samples and constructing an approximate problem that is tractable. It can be shown that under mild conditions the optimal solution (or set of solutions) and optimal value of the sampled problem converge to their deterministic counterparts as the sample size increases. CCP continues to attract the attention of researchers, and recent publications focused on other approaches such as the study of mixing sets ([2], [26]) and boolean functions [30], among others.

The class of problems that received less attention are the ones with pure binary variables, that is, $\mathcal{X} = \{0, 1\}^n$. In [3] and [42] the authors propose algorithms for specific versions of important binary problems. In [40] the authors propose an algorithm based on p -inefficiency points to solve several instances of the probabilistic set covering problem. The focus of this paper is on a new general algorithm for linear CCP with binary decision variables. The CCP can have either separated or joint chance-constraints, and in both cases the technology matrix and right hand-side vector can be random. The main idea is to use infeasible irreducible subsystems (IIS) to obtain cuts that can speed the

convergence of the algorithm. IIS were used in the context of CCP in [43], and the authors report significant reductions in terms of nodes explored and time with respect to the determinist equivalent formulation. However, the authors address the $\mathcal{X} = \mathbb{R}^n$ case with an individual chance constraint, and do not consider binary variables in the formulation.

Our methodology uses the functionalities of commercial solvers to generate IIS cuts. When the problem is infeasible with all scenario constraints included, the IIS identifies the source of infeasibility and indicates the constraints that should be removed to achieve feasibility. Interestingly, our methodology also addresses the feasible case, that is, even if a solution is available when all scenario constraints are considered, IIS can help find better solutions by removing some scenarios. For the feasible case, we introduce an additional constraint on the objective function value, and look for solutions that improve on the best upper bound obtained so far. In addition, our approach uses the fact that since $\mathcal{X} = \{0, 1\}^n$, we are able to estimate the minimal improvement that can be achieved in terms of the objective function value at each step. Through extensive computational experiments performed on separated and joint chance-constrained problems, we show that our approach can significantly decrease the number of nodes explored when compared to solving the problem directly using a commercial solver.

The rest of the paper is organized as follows. We start by defining IIS, providing examples and building the connection with CCP. In Section 3.3 we describe the formulations we will be working with, and the results we need in order to establish the validity of our approach. In Section 3.4 we describe the algorithm and discuss some implementation aspects. Section 3.5 presents the numerical experiments, and finally Section 3.6 concludes the paper.

3.2 Infeasible irreducible subsystems

In this section we give a brief introduction on IIS, and discuss how this idea can be applied to chance-constrained problems.

3.2.1 The deterministic equivalent formulation

In this work, we focus on chance-constrained problems with linear objective function and linear constraints. When the distribution of the random parameters is finite and belongs to set Ω , with $|\Omega| = S$, problem (3.1.1) can be written

in the following deterministic equivalent formulation:

$$\begin{aligned}
\text{DEP: } \min \quad & c'x \\
\text{s.t. } \quad & Ax \geq b, \\
& T(\omega)x + Mz_\omega \geq h(\omega), \quad \forall \omega \in \Omega, \\
& \sum_{s \in \Omega} p_\omega z_\omega \leq \alpha, \\
& z_\omega \in \{0, 1\}^S, \quad x \in \mathcal{X},
\end{aligned} \tag{3.2.2}$$

where p_ω is the probability of each element in Ω , and M is a constant that guarantees feasibility whenever z_ω is equal to one. Problem (3.2.2) can also be used when the distribution of the random parameters is continuous, or discrete with an infinite number of possible realizations. In those cases, a sample of size S is generated and formulation (3.2.2) can be regarded as an instance of a SAA problem. It can be shown (e.g. [32]) that optimal solutions and the optimal value converge to their exact counterparts. Guidelines can be found in [10] and [11] as to how to choose the sample size S such that with arbitrarily high probability the solution obtained in the SAA instance is feasible to the original problem.

For simplicity, we refer to formulation (3.2.2) as the deterministic equivalent formulation (DEP). For moderate sizes of S , the DEP can be solved directly; it is often the case that such approach is not efficient, mostly because the formulation contains the so-called big- M constraint, and relaxations tend to be very poor. The focus of this work is to solve the DEP efficiently when $\mathcal{X}$ is binary, and in what follows we will discuss how IIS can help on this task.

3.2.2 Irreducibly infeasible subsystems

Given a set of constraints Q , an IIS is a subset $P \subseteq Q$ such that

- a) It is infeasible;
- b) *Any* constraint p that is removed from P turns the resulting subsystem $P \setminus \{p\}$ feasible.

Example: Consider the following set of constraints Q (Figure 3.1):

$$\begin{aligned}
Q = \{ & x_1 - x_2 \leq 0, & \text{(A)} \\
& 2x_2 \leq 1, & \text{(B)} \\
& x_1 + x_2 \geq 2, & \text{(C)} \\
& x_2 \geq 1, & \text{(D)} \\
& 2x_1 + x_2 \geq 3\}. & \text{(E)}
\end{aligned}$$

It is easy to check that the following sets are IIS with respect to Q :

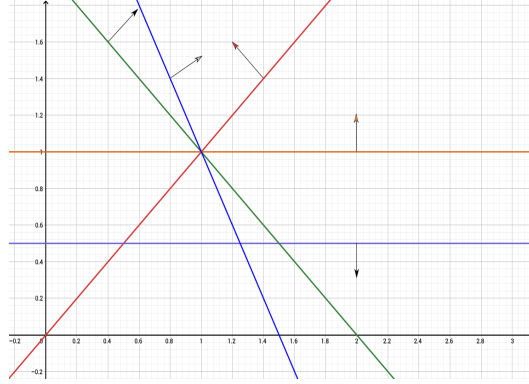


Figure 3.1: The set Q .

$$\begin{aligned} P_1 &= \{(A), (B), (C)\}, \\ P_2 &= \{(A), (B), (E)\}, \\ P_3 &= \{(B), (D)\}. \end{aligned}$$

In the continuous case with linear constraints, the work [21] provides a method to identify IIS that amounts to solve a linear problem. For other classes of problems, e.g. a nonlinear system of inequalities, specialized methods and approximations exist, and we refer the reader to the monograph [13] for details.

3.2.3 IIS and CCP Problems

In [43], in order to strengthen the DEP formulation (3.2.2) the authors generated cuts based on IIS. In their paper, the set $\mathcal{X}$ was continuous and they used the results of [21] to derive the cuts. To understand their approach, let us consider that the set Q of all S scenario constraints form an infeasible system. Of course this does not mean that the problem itself is infeasible: by removing scenarios such that sum of the respective probabilities remains less or equal to α , the problem must become feasible at some point (otherwise it was infeasible originally). The difficulty lies in choosing the correct constraints to be removed, that is, the constraints that lead both to feasibility and to the maximization of the objective function.

The first step is to consider a simplified version of problem (3.2.2), called the pure scenario problem (PSP), obtained by removing the knapsack constraint in problem (3.2.2):

$$\begin{aligned} \text{PSP: } \min \quad & c'x \\ \text{s.t. } \quad & Ax \geq b, \\ & T(\omega)x \geq h(\omega), \quad \forall \omega \in \Omega, \\ & x \in \mathcal{X}. \end{aligned} \tag{3.2.4}$$

If PSP is infeasible, by using [21] one can find an IIS, which naturally generates a cut for the DEP. For instance if constraints corresponding to scenarios 1, 4 and 7 form an IIS, the corresponding constraint, or cut, to be added to the DEP is

$$z_1 + z_4 + z_7 \geq 1.$$

Such constraint forces one of the scenarios that belong to this IIS to be left out of the problem. In [43] the authors propose a branch-and-cut algorithm on the variables z that searches for the best scenarios to be removed in the CCP. They report excellent computational results for an extensive set of instances.

It is not necessary to have an infeasible set of scenario constraints to justify the use of chance constraints. It may well be the case that the problem is feasible if all scenarios are included, and the purpose of the chance constraint is to improve the objective function value by removing some scenarios. In this situation, which refer to as the feasible case, it is not clear how IIS can be useful, since there is no infeasibility in the set of constraints. However, when we modify the PSP to include a constraint in the objective function value, infeasibility appears as a way of improving the quality of the solution. In other words, the question in this context is which constraints should be removed in order to improve the objective function value by at least some pre-specified amount. We refer to this formulation as PSPwO, and it can be written as follows:

$$\begin{aligned} \text{PSPwO: } \min \quad & c'x \\ \text{s.t. } \quad & Ax \geq b, \\ & T(\omega)x \geq h(\omega), \quad \forall \omega \in \Omega, \\ & c'x \leq u - \epsilon, \\ & z_\omega \in \{0, 1\}^S, \quad x \in \mathcal{X}, \end{aligned} \tag{3.2.5}$$

where u is an upper bound on the objective function value, obtained for example when solving the problem with all scenarios, and ϵ is an arbitrary value that defines the desired decrease in terms of objective function value. If the problem is infeasible it means that it is not possible to improve on the current bound *with all scenarios in place*. In this case an IIS is found, which necessarily contains the last constraint in PSPwO, and also other scenario constraints. As in the PSP problem, a cut can be constructed such that at least one of the constraints stopping the improvement in terms of objective function value has to be removed.

As we will see later, the choice of ϵ is extremely important in the performance of the algorithm. It is tempting to choose large values of ϵ in order to obtain quick improvements, but if the desired decrease is too large one may have to remove more constraints than allowed by α , which works as a budget on how many constraints can be removed. On the other hand, small values of ϵ may be too conservative, slowing down the computations. In [43] the authors do not report the value they worked with, and no experiments are performed comparing the performance of different choices. We will discuss in detail those issues in the

numerical experiments, and propose a scheme to choose the optimal ϵ in some sense.

For the binary case, that is, when $\mathcal{X} = \{0, 1\}^N$, we are not aware of a result as simple as the one devised in [21] to identify IIS. It is not the purpose of this work to propose a method or an heuristic to obtain IIS for binary problems, so whenever we need to generate one we use in-built functions of commercial solvers for that purpose. We tried to obtain more details of how those functions work, but there is very little information in the solvers' websites. Therefore, we use the functions as blackboxes that generate IIS whenever needed.

3.3 Strategies to solve binary CCP

In the previous section we described IIS and how they can be used to solve CCP, following the ideas described in [43]. In addition, by using in-built functions of commercial solvers one can in principle address the binary case. In this section we describe two fundamental ideas that make use of the structure of CCP, and of the fact that we are dealing with binary variables.

3.3.1 Efficient Upper Bound Generation

In every algorithm it is important to find upper bounds mainly for two reasons: to keep track of the best candidate as the algorithm advances, and to have a stopping criteria for termination. It is often the case that obtaining an upper bound is time consuming, and computing it at every iteration can slow down the algorithm significantly. Whenever the problem was feasible in [43], the authors used the PSP problem to obtain upper bounds. In the case where the PSP is feasible, the problem remains feasible as scenarios are removed. In this case, *as long as the sum of the probabilities of the scenarios removed is less or equal to α* , the solution of the PSP problem without those scenarios is feasible to the original problem (3.1.1).

We tried a similar approach for the binary case and the results were not satisfactory. The reason is that in our case PSP is a binary problem, and it is more demanding to solve than its continuous counterpart. In most cases, the time taken to compute upper bounds represented as much as 80% of computation time, which is not desirable for larger instances. We simply could not compute upper bounds as often as in the continuous case, but we still needed good upper bounds for fathoming nodes in our algorithm, and for closing the gap quickly. Our approach to deal with this problem is based on the following proposition:

Proposition 1. *Consider subsets P and Q of Ω such that $P \subset Q \subset \Omega$, and that*

$$\sum_{\omega \in \Omega \setminus P} p_s \leq \alpha,$$

that is, the sum of the probabilities of the elements that do not belong to P is less or equal to the reliability level α . For a set $A \subseteq \Omega$, let x_A be an optimal

solution and v_A the optimal value of problem (3.2.4) restricted to constraints defined by A . Assuming PSP is feasible when all scenarios are present, we have that

1. The solution x_Ω of PSP with all scenarios, is feasible for the original problem, and the optimal value v_Ω is an upper bound for the true optimal value v^* .
2. The solutions x_P and x_Q are feasible to CCP, and therefore the optimal values v_P and v_Q are upper bounds for v^* .
3. The optimal values of the original problem and of the PSP problem based on sets P, Q and Ω satisfy

$$v^* \leq v_P \leq v_Q \leq v_\Omega.$$

Proof.

1. As PSP is feasible when all scenarios are present, the optimal solution x_Ω of PSP is feasible to the original problem (3.2.2) when we fix $z_\omega = 0, \forall \omega \in \Omega$. Therefore $v_\Omega \geq v^*$.
2. For the set P , the optimal solution x_P of the PSP problem with P scenarios could violate the remaining scenarios constraints. However, by hypothesis the sum of the probability of scenarios belonging to $\Omega \setminus P$ is less or equal to α . Therefore, by fixing $z_\omega = 1$ for those scenarios we have a feasible solution to problem (3.2.2). The proof for Q is immediate since $P \subset Q$.
3. Since $P \subset Q \subset \Omega$, the result follows from 1 and 2.

□

Inspired by Proposition 1, our idea is to compute upper bounds only when the *budget* α is fully used, that is, only when one has removed as many scenarios as possible from PSP. For instance, suppose we have 100 scenarios, each one with the same probability, and $\alpha = .05$. This means five scenarios can be removed. We will only solve PSP when five constraints are removed from the set Ω : upper bounds obtained with a smaller number of constraints removed are likely to be worse than the ones obtained when the budget is completely used.

3.3.2 The choice of ϵ for binary problems

In the feasible case, in order to make use of IIS we add a constraint to PSP that accounts for solution quality. When the problem is feasible, the difficulty is not in finding a feasible solution after a constraint is removed, since any removal will generate a feasible solution because the feasible set is being enlarged. The problem is to choose the constraints to be removed such that the objective function value will decrease.

The value of ϵ should be such that all vectors which generate a decrease of *at least* ϵ in the objective function should be considered. In principle, one would have to use a very small ϵ , e.g. 10^{-5} , in order to avoid excluding candidates that may be optimal for the original problem. However, such small values of ϵ slow the algorithm down considerably, as improvements in the upper bound at every step are marginal.

In the discrete case more can be said about the choice of ϵ . The following proposition assures the existence of an optimal ϵ :

Proposition 2. *Let x^* be an optimal solution of problem (3.2.2), and $\bar{x}$ be a feasible solution for the problem such that $u = c'\bar{x} > c'x^*$. Then, there exists $\epsilon > 0$ that does not eliminate any vectors from problem (3.2.5) that improve the objective function value and satisfy $Ax \geq b$.*

Proof.

Let $\mathcal{W} = \{x | Ax \geq b, c'x < u\}$. By hypothesis, the set $\mathcal{W}$ is nonempty: the optimal solution $x^* \in \mathcal{W}$. We define ϵ as follows:

$$\epsilon = \min_{x \in \mathcal{W}} \max\{u - c'x, 0\}. \quad (3.3.6)$$

Since our feasible set is finite (and $\mathcal{W}$ is nonempty), the minimum in problem (3.3.6) will be attained, leading to a positive value of ϵ . Moreover, by the construction of the set $\mathcal{W}$, all vectors that improve on the value of u and satisfy $Ax \geq b$ are considered. $\square$

Unlike the continuous case, Proposition 2 gives a constructive way of finding a value of ϵ that guarantees a decrease in terms of objective function value *without excluding any potential candidate*. However, the result is not practical to implement since we would have to enumerate all vectors x that satisfy $Ax \geq b$. Nevertheless, inspired by the constructive proof of Proposition 2, we propose solving an alternative problem to find a value of ϵ :

$$\begin{aligned} \min_{\epsilon, x} \quad & \epsilon \\ \text{s.t.} \quad & \epsilon \geq u - c'x, \\ & \epsilon \geq 0, \\ & u - c'x \geq \delta, \\ & Ax \geq b, \\ & x \in \{0, 1\}^n, \end{aligned} \quad (3.3.7)$$

where $\delta > 0$ is a small enough number that serves the purpose of forcing the problem to move away from the current upper bound. This formulation allows us to find the smallest improvement (that is greater than δ) over the current upper bound u without having to enumerate all possible solutions. We will see that in practice the value of ϵ obtained by solving problem (3.3.7) provides excellent guidance for our numerical scheme.

In formulation (3.3.7) we are exploiting the fact that our decision variables x are binary. If the decision variables were integers problem (3.3.7) becomes challenging and it would slow down our numerical procedure significantly. Since binary variables are bounded, off-the-shelf solvers can quickly find a solution to problem (3.3.7).

3.4 The algorithm

In this section, we present our numerical scheme that uses IIS to solve binary CCP. First, we will define some auxiliary formulations that correspond to intermediate steps in the procedure, and then we present the step-by-step description of the algorithm. Finally, we show a proof of convergence.

3.4.1 Preliminaries

Let $\mathcal{N}$ be the set of open nodes indexed by k in a branch-and-bound (BAB) tree on the z_ω variables, $\omega \in \Omega$. Let a path from node k to the root node of the BAB tree be denoted by $\tau(k)$. We define $\mathcal{U}_k$ as the set of scenarios associated with nodes in $\tau(k)$ such that $z_\omega = 1$, and $\mathbb{P}(\mathcal{U}^k) \leq \alpha$. Similarly, define $\mathcal{L}^k \subseteq \Omega$ as the set of scenarios associated with nodes in $\tau(k)$ such that $z_\omega = 0$. The following formulation is similar the PSP problem, but it is restricted to scenarios that have not been removed yet:

$$\begin{aligned} \text{PSP2}^k: \quad & \min c'x \\ & \text{s.t. } Ax \geq b, \\ & T(\omega)x \geq h(\omega), \quad \forall \omega \in \Omega \setminus \mathcal{U}^k, \\ & x \in \mathbb{B}^n. \end{aligned} \tag{3.4.8}$$

As stated in §3.2.3, if (3.4.8) is feasible for $\mathcal{U}^k = \emptyset$ (i.e. the root node) then we can add to PSP2^k the constraint $c'x \leq u - \epsilon$, where u is the current upper bound. Now let S_j be an IIS of (3.4.8) and $D_j := \{\omega \in \Omega : T(\omega)x \geq h(\omega) \cap S_j \neq \emptyset\}$. If (3.4.8) is infeasible, we can obtain an IIS S_j that determines the set D_j . Then the IIS inequality $\sum_{\omega \in D_j} z_\omega \geq 1$ is valid for (3.2.2) by construction. Given the sets $\mathcal{U}^k$ and $\mathcal{L}^k$, the problem to solve at node k is:

$$\begin{aligned}
\text{PSP3}^k: \quad & \min c'x \\
\text{s.t.} \quad & Ax \geq b, \\
& T(\omega)x + Mz_\omega \geq h(\omega), \quad \forall \omega \in \Omega \setminus \{\mathcal{U}^k \cup \mathcal{L}^k\}, \\
& T(\omega)x \geq h(\omega), \quad \forall \omega \in \mathcal{L}^k, \\
& \sum_{\omega \in \Omega \setminus \{\mathcal{U}^k \cup \mathcal{L}^k\}} p_\omega z_\omega \leq \alpha - \sum_{\omega \in \mathcal{U}^k} p_\omega z_\omega, \tag{3.4.9a} \\
& \sum_{\omega \in D_j} z_\omega \geq 1, \forall j \in \tau(k), \tag{3.4.9b} \\
& x \in \mathbb{B}^n, z \in \mathbb{B}^S.
\end{aligned}$$

By adding an IIS inequality (3.4.9b) at a particular node of the BAB tree, at least one scenario is *excluded* from the nodal problem so that the total number of nodes to search in the BAB tree is reduced. Moreover, we note that constraint (3.4.9a) is updated based on scenarios already removed in the scenario tree, that is, belonging to the set $\mathcal{U}^k$.

3.4.2 IIS branch-and-cut algorithm

A detailed description is given in Algorithm 1.

Algorithm 1 IIS branch-and-cut algorithm (IISBAC)

- 1: **Initialize:** Set $\mathcal{L}^1 = \emptyset, \mathcal{U}^1 = \emptyset, n^1 = (\mathcal{L}^1, \mathcal{U}^1), \mathcal{N}^1 = \{n^1\}, K = 1, u = \infty, l = -\infty$.
 - 2: **Node Choice:** Pick some node $n^k \in \mathcal{N}$ according to some search rule. If $\mathcal{N} = \emptyset$, then terminate execution.
 - 3: **Solve LP:** Solve (3.4.9) relaxing the integrality on the z variables. This will either find an optimal solution $(\bar{x}, \{\bar{z}_\omega\}_{\omega \in \Omega})$, or that the problem is *infeasible*.
 - 4: **Fathoming Rule:** If the node relaxation is infeasible, or $c'x \geq u$, then fathom the node and return to 1. Otherwise, solve (3.4.8). If (3.4.8) is feasible, then set $z_\omega = 0, \forall \omega \in \Omega \setminus \mathcal{U}^k$ obtaining a feasible integer incumbent, update the upper bound and go to 2. Otherwise go to 5.
 - 5: **IIS Cut Generation:** Obtain the set D_j using some IIS search method. If $D_j \neq \emptyset$, then add the cut $\sum_{\omega \in D_j} z_\omega \geq 1$ and go to 3. Otherwise, improve (if possible) the upper bound u and go to 6.
 - 6: **Branching:** Pick a non-integer z variable using some branching rule. Create new nodes $n^{k+1} = (\mathcal{L}^k \cup z_\omega, \mathcal{U}^k)$ and $n^{k+2} = (\mathcal{L}^k, \mathcal{U}^k \cup z_\omega)$. Add these nodes to $\mathcal{N}$, set $k = k + 2$ and go to 3.
-

In our implementation, we used Proposition 1 to improve the algorithm in step 4. Given a set $\mathcal{U}^k$, we define $Q = \Omega \setminus \mathcal{U}^k$. By removing additional scenarios

from Ω and adding them to set $\mathcal{U}^k$ such that the budget α is not exceeded, we construct the set $\mathcal{U}^{k'} \supset \mathcal{U}^k$, and define $P = \Omega \setminus \mathcal{U}^{k'} \subset Q$. Following the third claim of Proposition 1, the upper bound obtained using the set P is less than or equal to the one based on set Q . It is important to highlight that such scheme is only valid for the computation of the upper bound: the cut is always constructed based on the original set $\mathcal{U}^k$.

Our algorithm works like a standard branch-and-cut algorithm, as we are only adding an intermediary step in order to add our cuts using the information obtained by the IIS. This algorithm still relies on finding lower bound (improved by our cuts) and upper bounds (improved by our checking step, when possible), and closing the gap between them as the termination criterion.

Fathoming nodes is important to speed up the process of finding the optimal solution. We have two rules: fathoming by optimality or infeasibility. The former can be applied by checking the upper bound (step 2), the latter by not being able to find an IIS that relies on stochastic constraints (i.e. all IIS members are rows of the A matrix). Termination is guaranteed by the following proposition.

Proposition 3. *The IIS branch-and-cut algorithm terminates in a finite number of iterations and finds the optimal solution if it exists.*

Proof. In order to prove finiteness, consider we are solving our problem (3.2.2) with $n + S$ binary variables, then we have 2^{n+S} possible combinations of solutions, thus in the worst case we might explore all possible solutions, which are finite.

Now, we must prove that the algorithm does not remove the optimal solution. As we have discussed before: there are two main cases to consider. If (3.2.4) is infeasible in the root node, and assuming (3.2.2) has an optimal solution, the algorithm focuses in cutting infeasible leaves, which are a subset of the 2^{n+S} possible combinations. Therefore, we are reducing the number of combinations to explore, while avoiding cutting the optimal solution. Since we are solving (3.4.9), we are always checking feasibility in the knapsack, then the solution will be feasible (3.2.2).

There is another case to consider: if problem (3.2.4) is feasible at the root node. While we select an $\epsilon > 0$, the algorithm cuts feasible leaves of solution combinations and by construction the upper bound improves each time, since we will fathom all nodes that have a worst bound than the incumbent upper bound. This means we are not cutting nodes which might contain the optimal solution, since they have solutions which yield values strictly smaller than our current upper bound. Henceforth, we are reducing the number of nodes to explore. $\square$

As mentioned, obtaining an IIS for binary problems is involved, and we use function `ComputeIIS()` of the `model` class provided by the solver Gurobi. As mentioned, our focus is not on finding a method to obtain an IIS with binary variables, instead we concentrate on obtaining valid cuts using an IIS. Unfortunately, there is no documentation about *how* this function works, and we have no control over it.

3.5 Numerical experiments

We present two sets of experiments: first, consider a joint chance-constrained version of the probabilistic set-covering (PSC) problem, as described in [7]. In the problem matrix T has dimensions $m_2 \times n$, $m_2 > 1$. This means that we have a set of m_2 constraints inside the chance-constraint. The only source of randomness in this problem is vector $h(\omega)$, with dimensions $m_2 \times 1$. Furthermore, the coefficients of the objective function and the entries of matrix T are binary. As we will show, we need to pre-process the problem in order to solve it efficiently. In addition, writing the cut derived by the IIS is more challenging in this situation because we have joint chance-constraints. The z_ω variables have a different meaning in this case: whenever z_ω is equal to one, a whole block of m_2 constraints is removed, rather than a single constraint. Since the IIS oracle returns constraints, it is not clear how to move from constraints to blocks for IIS cut construction. We propose a rule that tells us how many and which blocks to remove given the set of constraints belonging to the IIS, and tested our approach on several instances of the PSC problem.

The second experiment is a vaccine allocation problem, where $T(\omega)$ is a $1 \times n$ matrix and $h(\omega) \in \mathbb{R}$, for all $\omega \in \Omega$. In this case, the application of IIS cuts is direct: whenever a constraint belongs to an IIS, it is added as an element in the cut. The coefficients in the objective function are non-integers, and we will show that it is very important in this case to choose ϵ appropriately. This problem was analyzed in [43] for continuous variables.

The objective of our experiments was to validate the methodology and to reduce the number nodes explored in the IISBAC algorithm compared to a standard solver. In this sense, we want to compare number of nodes explored, which does not depend on the programming language used to implement the algorithm. For the sake of completeness, we also compare running times.

In our experiments we used an implementation of IISBAC algorithm in Python 3.6.2 using Gurobi 7.5.1. The experiments were run on an Intel(R) Xeon(R) CPU E5-2670 @ 2.60GHz (using 8 threads), 32 GB of memory and running CentOS 6.8.

3.5.1 Stochastic Set Covering Problem

The formulation for this problem as described in [7] is given as follows:

$$\begin{aligned}
 \text{PSC: } \min \quad & c'x \\
 \text{s.t. } \quad & Tx + z_\omega \geq h(\omega), \quad \forall \omega \in \Omega, \\
 & \sum_{\omega \in \Omega} z_\omega \leq \lfloor S\alpha \rfloor, \\
 & z_\omega \in \mathbb{B}^S, \quad x \in \mathbb{B}^n.
 \end{aligned} \tag{3.5.10}$$

Problem (3.5.10) is a joint chance-constrained problem, so each scenario ω corresponds to a block of constraints. The difficulty here is that the IIS routine returns the constraints to be removed, while we need to decide on the blocks to

be removed. The cuts based on IIS would still be valid, and would eventually reach the optimal solution. However, we could not reach the optimal solution for most instances. We realized that in this particular case, we could use a heuristic: let the *cut length* l be the amount of blocks that will belong to the cut (i.e. the amount of z_ω variables present in the cut). So, by bounding the cut length to a fixed number, we will only include the first l blocks selected by sorting them in nondecreasing order of the number of *constraints* that are part of the IIS, which is the output given by the solver. For instance assume $l = 2$ and we have three blocks of constraints, each with eight constraints ordered accordingly. If constraints 1, 9, 12, 15, 18 and 22 are part of an IIS, we would add the cut $z_2 + z_3 \geq 1$ because the first block has only one constraint belonging to the IIS (constraint 1), and blocks 2 and 3 have three (9, 12 and 15) and two constraints (18 and 22), respectively.

Results

There was a preprocess step used to reduce the number of stochastic constraints in this problem. We use $\alpha = 0.1$ for this problem, and for instance in the case of 1,000 samples our removal budget is equal to 100. Since only the right hand side is random, whenever we have more than 100 copies of a constraint the budget is not enough to eliminate of such constraint. In this case, since the constraint will have to be satisfied, we added it as a row of the system $Ax \geq b$. Using this simple technique, which surprisingly was not done automatically by Gurobi, allowed the solver to handle the instances considered in the experiment.

Table 3.1 shows the performance of both Gurobi and IISBAC using different cut lengths for 200 replications, with sample sizes ranging from 100 until 10,000. The mean value of ϵ was 1, with a variance at most $O(10^{-10})$ among the replications. To understand the effect of the cut length parameter l , we used values between 1 and 50. It can be seen that as l increases, the values obtained by IISBAC converge to the optimal value found by Gurobi. This requires exploring a larger number of nodes, and taking more time on average. The PSC `scp41` from [6] is a joint chance-constrained problem with blocks of size 200, which makes it extremely challenging to solve it. Our main purpose was to show that IIS can handle this problem if l is sufficiently large, and that the computational times and nodes explored remain in the same order of magnitude as Gurobi's.

We believe significant time reductions can be achieved by using more sophisticated implementations of the IISBAC. Regarding nodes explored, Gurobi identifies the structure of the PSC problem and uses sophisticated cuts in order to explore less nodes. It seems that for other joint chance-constrained problems IIS cuts could have a significant impact, and could reduce the total numbers explored by a significant amount. The heuristic shows promising results for the joint chance-constrained case, but further research is needed to fully test more criteria in terms of choosing a cut length, or the prioritization to select which scenarios to choose for the cut. Our findings show that if optimality is not critical, very good solutions (within 1% of the optimal) can be found quicker than Gurobi by using smaller cut lengths. By finding good candidates quicker, we

Sample Size	Average	Gurobi	<i>IISBAC Cut Length l</i>					
			1	2	5	10	20	50
100	Nodes	2.5	2.7	2.8	2.7	2.7	2.8	4.3
	Obj. Val.	377.0	378.4	378.1	377.8	377.6	377.3	377.0
	Cuts	-	0.1	0.1	0.2	0.2	0.4	0.0
	Time (s)	0.1	6.2	7.8	9.5	13.7	18.2	32.9
1,000	Nodes	1542.3	927.0	3092.0	3142.5	3551.7	2661.0	2089.6
	Obj. Val.	382.9	383.8	383.4	383.1	383.0	383.1	382.9
	Cuts	-	3.5	4.6	5.3	6.2	6.9	0.0
	Time (s)	17.0	50.2	101.2	100.4	111.4	109.2	130.3
5,000	Nodes	921.0	1043.9	1937.9	1401.7	1537.3	1774.7	1254.0
	Obj. Val.	381.1	381.3	381.2	381.2	381.1	381.1	381.1
	Cuts	-	2.7	2.8	3.6	3.4	3.0	0.0
	Time (s)	168.5	247.8	380.1	344.4	398.4	393.2	264.9
10,000	Nodes	3.3	8.7	6.5	11.8	12.4	11.6	7.6
	Obj. Val.	380.1	380.2	380.1	380.1	380.1	380.1	380.1
	Cuts	-	2.3	2.4	3.2	3.3	3.3	0.0
	Time (s)	90.5	144.2	146.1	156.9	163.1	176.0	176.5

Table 3.1: Comparison of performance between Gurobi and IISBAC using different sample sizes in `scp41`.

could use this approach to give the solver a good starting point. If optimality is critical, then larger cut lengths are needed, possibly combined with another cutting generation procedure to speed up the method.

In order to gain a deeper understanding of our approach, we also applied this methodology for PSC problems `scpe1` (blocks of size 50) and `scpa1` (blocks of size 300). For `scpe1` IISBAC found the optimal solution when using 1,000, 5,000 and 10,000 samples in less than 600 seconds. Moreover, for those smaller instances the computations were not sensitive to the cut length, and the algorithm visited less nodes than Gurobi.

Table 3.2 describes the results for `scpa1`, and in this case a time limit of 7,200 seconds was imposed given the size of the problem. Results show that IISBAC obtained solutions within 2% of the optimal, and the number of nodes explored is smaller than Gurobi's. The effect is more salient as the sample size increases, showing the potential of IISBAC for solving linear large-scale joint chance-constrained problems.

Sample Size	Average	Gurobi	<i>IISBAC Cut Length l</i>					
			1	2	5	10	20	50
1,000	Nodes	622.54	436.97	396.94	417.72	449.81	467.06	454.87
	Obj. Val.	232.16	233.89	233.74	233.54	233.34	233.24	233.05
	Cuts	-	4.34	4.29	4.53	4.26	4.69	4.94
	Time (s)	44.09	6345.88	6486.21	6321.03	5545.52	6174.03	5928.11
5,000	Nodes	346.14	83.63	84.075	84.37	85.25	86.4	89.16
	Obj. Val.	232.4	234.43	234.28	234.18	234.04	233.87	233.68
	Cuts	-	4.34	4.22	4.69	4.07	4.54	4.32
	Time (s)	725.01	6379.05	6450.75	6998.06	6754.95	6844.34	6919.18

Table 3.2: Comparison of performance between Gurobi and IISBAC using different sample sizes in `scpa1`.

3.5.2 Vaccine Allocation Problem

The formulation for this problem is:

$$\begin{aligned}
\text{VAC: } \min \quad & c'x \\
\text{s.t. } \quad & Ax = 1, \\
& T(\omega)x + M_\omega z_\omega \geq 1, \quad \forall \omega \in \Omega, \\
& \sum_{s \in \Omega} z_\omega \leq \lfloor S\alpha \rfloor, \\
& z_\omega \in \mathbb{B}^S, \quad x \in \mathbb{B}^n.
\end{aligned} \tag{3.5.11}$$

The objective is to minimize the cost of assigning vaccines between families in a given sector (given by the A matrix), while bounding the probability of spreading the contagion between these families (given by the T matrix). In this case, it was needed to use a SAA approach to obtain samples of size S from the random variable. As stated before, for every $\omega \in \Omega$ we have that $T(\omega)$ represents just one constraint.

Table 3.3 presents the problem sizes for the test instances. As can be seen, the number of decision variables stays constant, equal to $n = 302$, and the number of rows in the A matrix is fixed in 30. Nonetheless, the number of scenario variables (and rows) increases depending of the sample used. Therefore, S ranges from 100 until 2,000. As stated in [43], these instances are difficult to solve because the MIP formulations are large and dense. We included a last column which describes the number of scenarios that can be violated at the same time in an incumbent solution (we consistently used the value $\alpha = 0.05$).

Results

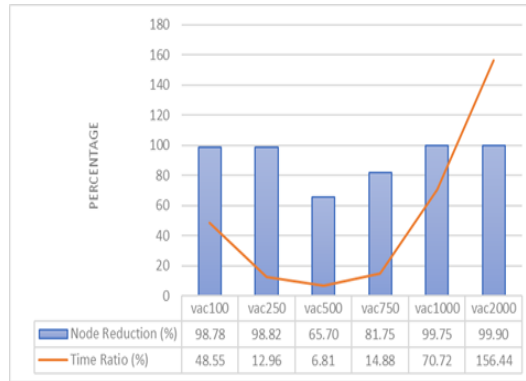
First we present a summary of our findings, and then we highlight some salient features of our approach. Figure 3.2 shows a comparison between the performance of Gurobi and IISBAC algorithm throughout the different instance sizes.

Instance	Rows	Decision vars.	Scenario vars.	Knapsack budget*
vac100	131	302	100	5
vac250	281	302	250	12
vac500	531	302	500	25
vac750	781	302	750	37
vac1000	1,031	302	1,000	50
vac2000	2,031	302	2,000	100

Table 3.3: Problem sizes for vaccination test instances. * $\alpha = 0.05$

The node reduction, defined as the ratio of nodes explored between Gurobi and IISBAC was on average 90.78%, and on larger instances it was over 99%. This is a remarkable difference with respect to the joint case, which indicates that individual constraints, or possible joint chance-constraints with moderate values of m_2 , could benefit from our approach.

In terms of time, Gurobi is able to outperform IISBAC on small instances, but the solution times are comparable when we used 1,000 samples. Even though we are using a language suitable for prototyping, our implementation was faster than Gurobi for the larger 2000-sample instances. Our explanation is that as the problem becomes larger, more effort is necessary to solve the problem as new constraints and binary variables are being added. Thus, as complexity rises, any help to reduce said effort is going to have a significant impact on the solver's performance.

**Figure 3.2:** Performance comparison between instances.

We ran more tests with different values of ϵ in this formulation, since it is easier to solve than the PSC. We were consistently lowering the number of nodes explored, which was our objective, but we noticed that the variance of the ϵ 's calculated by (3.3.7) was at most $O(10^{-2})$ with mean 0.3076. The practical conclusion is that problem (3.3.7) is essential to the performance of the algorithm, but it does not need to be solved more than once.

To stress this fact, we note that using different values of ϵ will produce very different outcomes: if $\epsilon < 0.3076$, IISBAC will produce smaller improvements on the upper and lower bounds, which in turn means more nodes to explore. We were able to detect that for values close to 0.001, the amount of nodes can quickly surpass Gurobi's count. On the other hand, if $\epsilon > 0.3076$ then the optimal solution could potentially be cut from the tree.

Table 3.4 zooms in on larger instances considered for this problem, consisting of 1,000 and 2,000 samples. On average, the IISBAC algorithm explored between 0.01% and 2% of the total nodes of Gurobi, with less computational time in half of the instances. We detected that the first node added was able to improve the upper bound obtained by Gurobi up to 0.1% over the optimal value in every case (between the first 3 minutes of running time), while the rest of the time was spent to close the gap by improving the lower bound via cuts.

<i>Sample</i>	<i>Gurobi</i> <i>IISBAC</i>		<i>Gurobi</i> <i>IISBAC</i>		<i>IISBAC Info</i>	
	Time		Nodes		Node Red. %	Cuts
vac1000a	2,936	4,344	144,904	2,910	97.99%	169
vac1000b	1,000	3,229	156,803	2,362	98.49%	213
vac1000c	3,449	10,045	1,056,888	1,275	99.88%	168
vac1000d	6,488	2,943	1,945,060	1,365	99.93%	176
vac1000e	2,437	2,503	546,318	1,536	99.72%	9
vac2000a	51,456	14,980	3,335,160	3,438	99.90%	255
vac2000b	30,218	83,717	1,540,281	5,827	99.62%	474
vac2000c	>100,000*	66,066	>12,184,026*	9,325	99.92%	661
vac2000d	84,524	24,043	4,555,042	3,626	99.92%	310
vac2000e	69,259	25,623	6,034,681	4,127	99.93%	292

Table 3.4: Larger samples (1,000 and 2,000) results using $\alpha = 0.05$. * final gap 1.5%.

Finally, the second half of Table 3.4 shows promising results. IISBAC explores less than 1% of the nodes that Gurobi explores on average, while doing so in less time than the solver: we can observe solving times being around 25% those of the solver. Note that instance *vac2000c* using Gurobi was only able to reach a gap of 1.5% after 600,000 seconds (almost 7 days), when it was decided to interrupt the computations.

As can be seen with the vaccine problem, the IISBAC algorithm is able to reduce the effort in solving this problem, and as the amount of samples used to solve the problem increases, so does the effectiveness of the IIS cuts applied. We were able to observe that while the solver works on the root node of the branch-and-bound tree, the first nodes added by IISBAC algorithm are able to obtain on average an upper bound within 1% of the optimal value, while the rest of the time was an effort to close the lower bound, using the cuts.

It is important to highlight that when α is greater than or equal to 0.1 the problem becomes more complex. The combinatorics for the possible ways of choosing which constraints can be violated increases, and both Gurobi and our algorithm provide unsatisfactory solutions. For $\alpha = .1$ Gurobi cannot reduce

the optimality gap to less than 5% after two hours in any of the 1,000 and 2,000 samples instances. The same happened with IISBAC: the gap was around 20% in most instances under the same time limit, while the number of nodes explored was around .02% of the nodes explored by Gurobi. In personal communications with a senior developer of Gurobi we concluded that this vaccine problem is especially challenging because some of the coefficients are of small magnitude small compared to the objective function value coefficients. We believe that for other linear binary chance-constrained problems IISBAC would perform better for values of α greater or equal to 10%.

3.6 Conclusions

In this paper, we propose a novel methodology to solve binary chance-constrained problems building on [43]. In particular, we develop improvements for finding upper and lower bounds by using the structure of the problem to our advantage. A salient feature of our approach is the idea of adding a constraint that can improve on the quality of the solution at every iteration, which allows us to handle cases where the problem with all scenarios in place is feasible. We solve an auxiliary problem that allows to define the maximum possible improvement at every iteration, and the efficiency depends directly on the fact that our decision vector is binary.

The algorithm can handle joint and individual linear CCP, and we tested our approach on a probabilistic version of the set covering problem, and on a vaccine allocation problem. In the first case, a joint CCP, some extra work is required to convert the output of the IIS generator to cuts that remove blocks of constraints, and a heuristic was implemented to handle the priority in which these blocks should be added. Computational results, on problems with 50, 200 and 300 constraints per scenario (we considered up to 10,000 scenarios) indicate that the method finds the optimal solution for cuts with length 50, and finds solutions within 1% of the optimal for smaller cut lengths, with less computational effort. For individual CCP computational results show that in many instances we reduced drastically the number of nodes explored, and in some cases the computational time was comparable to Gurobi even though we did not aim at maximum computational efficiency.

The main goal of the paper was to establish IIS as a tool to solve binary CCP. For the joint case more research is needed to reduce computational times, and for the individual case the results indicate that significant gains in terms of speed can be achieved using our methodology. A natural extension of the current work would start with problems with a single chance-constraint, and move gradually to more complex joint CCP, in order to gain a better understanding of the impact in performance of moving from constraint removal to block removal. For higher values of the reliability level α the results were unsatisfactory, both with Gurobi and our algorithm. We believe it is due to the small magnitude of some coefficients in the constraints with respect to the constants in the objective function. Further research is needed in order to be able to cope with the

combinatorial explosion that occurs when more constraints are allowed to be violated. On the applications side, binary CCP arise often in natural resource management problems, in particular in mining applications where the decision variables are whether or not to remove a block from the mine. Uncertainty is present via ore prices or grade distribution, and the objective is to maximize net present value of the operations, a description that fits into the framework developed in this work.

Chapter 4

The risk-averse ultimate pit problem

4.1 Introduction

A fundamental problem in open-pit mining is the determination of the ultimate pit (UP), which consists in finding the contour of the mine that maximizes the difference between profits obtained from minerals minus extraction costs. The problem is formulated on a representation of the mine into blocks of a certain size, taking into account engineering requirements such as slope and precedence constraints. The formulation of the UP problem can be traced back to 1965: the seminal paper of [31], which was one of the first to propose an algorithm to solve this problem, and their methodology has been widely used in the mining industry.

The UP problem is relevant in practice mainly for two reasons: first, it gives an estimate of the total mineral that can be potentially extracted from an economical point of view. Second, the solution allows one to generate the so-called *nested pits*: by solving the problem for different prices (or “revenue factors”) of the mineral it is possible to obtain a sequence of pits such that higher prices generate larger pits that contain the previous ones. Such nested pits are the input for heuristics capable of generating a long term order of extraction of different sectors of the mine. Moreover, they provide an estimation of the different sizes of the pit under less favorable price conditions.

The vast majority of work in the UP problem assumes the parameters of the problem are known. In particular, the ore distribution is inferred by drilling, using techniques from geostatistics, such as *kriging* [16]. Such estimates are usually an approximation, and determining the ultimate pit by replacing the ore distribution by a single number per block, such as the average grade, can have negative consequences. There are few works that incorporate the distribution of ore grades into the UP problem. In [33] the authors use expected profits in the objective function, and conclude that larger profits can be obtained by

incorporating uncertainty into the UP formulation with respect to the deterministic case, and that the relative gains of the stochastic approach increase with the treatment costs. In [19] the authors move one step further and study the risk-averse UP problem, replacing the expected value by risk measures such as the Conditional Value-at-Risk (CVaR) [39] which has been widely used in applications such as finance and energy.

We propose a detailed study of the risk-averse UP problem, assuming that the grade of each block follows a known distribution. First, we discuss desirable properties—risk-nestedness and additive consistency—that a risk measure should have in defining the UP, and show that the only risk measure that satisfies those properties is the entropic risk measure. In particular, the CVaR does not satisfy any of these properties and is probably not the best choice to capture risk-aversion in the mining context. Second, we derive conditions under which the entropic risk measure generates nested pits by varying the *risk-aversion level* of the decision maker.

We apply our methodology to a small case study of a self-constructed instance to illustrate the gains of our approach. We also validate methodology on a real-world mine. Furthermore, we show how ore grade uncertainty can drastically change the solutions obtained, while the current methodology remains stoic to the stochastic nature of the ore grade.

The structure of this paper is as follows: Section 4.2 is an overall view of the UP problem, and we present a two-stage stochastic formulation as well as summarize the concept of nested pits. In Section 4.3 we define risk in the UP problem and propose desirable properties risk measures should have. In Section 4.4 we show the results of our methodology using a small instance to visualize the differences between the our and the classic methodology, and using a real mine to show the potential of its use in real life instances. Section 4.5 concludes the work and points out futures avenues of research.

4.2 Ultimate Pit problem and nested pits

4.2.1 UP problem

Deterministic version

The UP problem consists in finding the last set of blocks to be extracted in order to maximize the value of the mine, respecting precedence constraints. It is assumed that capacity is unlimited and that the ore can be mined immediately, in the sense that time considerations are left out of the formulation. In other words, the UP solution gives the contour of the mine without specifying *when* the blocks will be extracted.

We will define the parameters which will be used in the paper.

- B : set of blocks of the mine.
- $P \subseteq B \times B$: set of precedences for every block in the mine, that is, if $(b, b') \in P$ then in order to extract block b we must extract block b' .

- c_b^e : cost of extracting block b .
- c_b^p : cost of processing block b .
- r_b : profit for processing block b in the case that its grade is 1.0.
- g_b : ore grade of block b .

Parameter r_b includes, among other factors, the tonnage of the block, the recovery of the metal after processing the block, and the price of the metal in the market. As a convention, $c^e := \{c_b^e\}_{b \in B}$ and $c^p := \{c_b^p\}_{b \in B}$ will be defined as the vector of all extracting costs and processing costs for every block $b \in B$, respectively.

We can formulate the UP problem as the following mixed-integer optimization problem:

$$\begin{aligned}
 \text{UP}(g, r) = \min_{x^e, x^p} \quad & \sum_{b \in B} (c_b^p - r_b g_b) x_b^p + c_b^e x_b^e \\
 \text{s.t.} \quad & x_b^e \leq x_{b'}^e, \forall (b, b') \in P, \\
 & x_b^p \leq x_b^e, \forall b \in B, \\
 & x_b^e, x_b^p \in \{0, 1\} \forall b \in B,
 \end{aligned} \tag{4.2.1}$$

where the decision of extracting and processing each block $b \in B$ is represented by the binary variables x_b^e and x_b^p respectively. Since (4.2.1) is a minimization problem we will assume costs as positive and benefits as negative values, a convention we will use throughout the paper. The first set of constraints represents the extraction precedents for every block and the second set of constraints condition the processing of a block to its extraction.

UP with uncertainty on ore grades

The first step to develop a risk-averse model is to define a way to handle ore grade uncertainty for each block. In practice, formulation (4.2.1) is commonly solved by assuming g_b is deterministic, usually the mean value of different scenarios given by a geostatistical model.

For each $b \in B$, let $\tilde{g}_b$ be the random variable that represents the ore grade of block b . With this modification, the UP problem under uncertainty can be formulated as a two-stage stochastic optimization problem in which first stage variables represent block extraction, and in the second stage we have processing decisions, given extraction decisions taken in the first stage problem and the observed ore grade of extracted blocks. The two-stage stochastic UP problem can be formulated as follows:

$$\begin{aligned}
 \text{UP}_U := \min_{x^e} \quad & \sum_{b \in B} c_b^e x_b^e + \rho_\alpha[Q(x^e, \tilde{g}, r)] \\
 \text{s.t.} \quad & x_b^e \leq x_{b'}^e, \forall (b, b') \in P, \\
 & x_b^e \in \{0, 1\} \forall b \in B,
 \end{aligned} \tag{4.2.2}$$

where

$$\begin{aligned} Q(x^e, \tilde{g}, r) := \min_{x^p} \quad & \sum_{b \in B} (c_b^p - r_b \tilde{g}_b) x_b^p \\ \text{s.t.} \quad & x_b^p \leq x_b^e \quad \forall b \in B, \\ & x_b^p \in \{0, 1\} \quad \forall b \in B. \end{aligned} \tag{4.2.3}$$

Function $\rho_\alpha : L^1 \rightarrow \mathbb{R}$ will be defined as a *deviation or risk measure*, which deals with the uncertainty of the ore grade in the second stage. The parameter $\alpha \in \mathbb{R}$, whose range depends on the risk measure ρ_α under consideration, represents the risk-aversion of the decision maker. The work [33] did an extensive investigation when $\rho_\alpha = \mathbb{E}$, showing the importance of using the stochastic UP problem versus the deterministic version, and how it affects the pits obtained.

4.2.2 Nested pits

Usually, after the UP is established, mine planners run a complete scheduling model that defines in which period each block, or clusters of blocks, will be extracted. To this end, they solve $\text{UP}(g, (1 - \beta)r)$ as in (4.2.1) where $\beta \in [0, 1[$ is a multiplier of the benefit r_b . Figure 4.1 shows the pits generated by solving a small instance of problem (4.2.1) with $\beta_1 > \beta_2 > \beta_3 > \beta_4$.

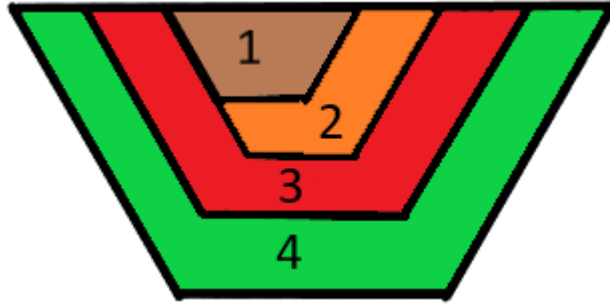


Figure 4.1: Example of nested pits.

As seen in the figure, the solution of $\text{UP}(g, (1 - \beta_i)r)$ will be a subset of blocks for any solution of $\text{UP}(g, (1 - \beta_{i+1})r)$ where $\beta_i > \beta_{i+1}$. This property is known as *nested pits*, and it is widely used in the mining context (for more information, please refer to [9, 23, 31]). These pits are an important input for scheduling heuristics to determine the future workload and planning effort in mining.

Such methodology can be seen as a simple heuristic to deal with ore and price uncertainty. Using larger values of β translates into smaller pits where it is *safe* to start working as the mineral found in that pit justifies the extracting and processing costs even in low price scenarios. However, since ore grade is assumed to be deterministic, this approach does not capture the effects of ore grade variability directly.

An important aspect in our work is to propose an adequate risk measure ρ_α for mining problems, and to be able to solve problem (4.2.2)-(4.2.3) efficiently. Uncertainty of the ore grade is taken into account, and we want to derive conditions under which nested pits are obtained by varying the parameter α .

4.3 Risk measurement

4.3.1 Defining risk under ore grade uncertainty

In order to select the appropriate risk measure for the UP problem, we need to define which are the desirable characteristics in the pits generated by a given risk measure ρ_α . First, it needs to yield nested pits by varying α : *conservative* pits must be contained in *riskier* pits. The ability to generate nested pits based on parameter variation should be considered as a mandatory property for a risk measure since mine exploitation scheduling uses nested pits as a main input. The precise definition of *risk nestedness* is as follows:

Definition 1. *Assuming as the level of risk-aversion of the decision maker rises then the value of $\alpha \in \mathbb{R}$ increases and the ore grades are independently distributed, a risk measure ρ_α is risk nested for the UP problem if for $\alpha_1 > \alpha_2$ the set of extracted blocks in the optimal solution of problem (4.2.2) obtained by using ρ_{α_1} is contained in the set obtained using ρ_{α_2} .*

The CVaR is a commonly used risk measure thanks to its tractability. Following [39], the CVaR can be defined as

$$\text{CVaR}_\alpha[X] = \min_{\eta \in \mathbb{R}} \left\{ \eta + \frac{1}{1-\alpha} \mathbb{E}(-X - \eta)_+ \right\}, \quad (4.3.4)$$

where X is a random variable and $(a)_+ := \max(a, 0)$. Remark that $\alpha \in [0, 1[$. A value of zero corresponds to the risk-neutral case and as α approaches one the risk measure protects against the worst-case realization. The next proposition shows that it may not be the ideal risk measure for stochastic UP problems under our requirements.

Proposition 4. *The CVaR risk measure is not risk-nested for the UP problem.*

Proof. We will prove that the CVaR is not risk-nested with a counter-example: let three blocks be positioned as in Figure 4.2.

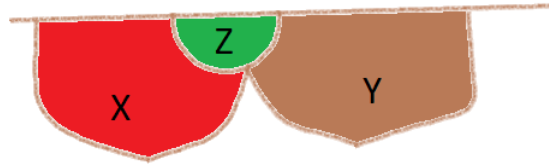


Figure 4.2: Three blocks.

In order to extract block X or Y , we must extract block Z (i.e. Z is a precedence of X and Y). Let the ore grade of the three blocks be independent of each other and normally distributed. If $W \sim N(\mu, \sigma^2)$ is normally distributed with mean μ and variance σ^2 , then

$$\text{CVaR}_\alpha(W) = \mu + \frac{\sigma^2}{(1-\alpha)\sqrt{2\pi}} e^{-z_{1-\alpha}^2/2}, \quad (4.3.5)$$

where $z_{1-\alpha}$ is the $(1-\alpha)$ -quantile of the standard normal distribution. Let $\tilde{X} \sim N(-2, 18)$, $\tilde{Y} \sim N(1, 0)$ and $\tilde{Z} \sim N(-4, 10)$ be the random variables that represent the negative profit of each block.

Table 4.1 shows the results of computing the CVaR using (4.3.5) for different values of α . It can be seen that with $\alpha = 0.05$ it is optimal to extract blocks X and Z , but with $\alpha = 0.1$ the optimal solution is to extract Y and Z . Finally with $\alpha = 0.5$ the preferences are such that it is better not to extract any blocks. Since $0.1 > 0.01$, extracting Y and Z and then X and Z shows that the CVaR is not risk nested. $\square$

Combination of blocks	α		
	0.05	0.1	0.5
Nothing	0	0	0
$X + Y + Z$	-1.96	0.46	17.34
$X + Z$	-2.96	-0.54	16.34
$Y + Z$	-1.91	-1.05	4.98

Table 4.1: Comparison of mining preference under CVaR.

A second desirable property is related to the consistency of extracting decisions. Current geostatistical techniques assume a dependence over ore grades within blocks of the mine that are physically near, but this dependence diminishes as we select blocks that are far from each other. In other words, assuming two blocks of the mine could be represented as X and Y , the decision of mining X instead of Y should not depend whether we are now considering another block Z which could be far from blocks X and Y , or of a different geological composition, acting as an independent random variable. In summary, we believe it is desirable that a risk-measure used in mining satisfies the *additive consistency* property:

Definition 2. Let X, Y and Z be random variables where Z is independent of both X and Y , and $\rho_\alpha(\cdot)$ be a risk measure where α is the risk level of the decision maker. If $\rho_\alpha(X) < \rho_\alpha(Y)$ and ρ is additive consistent, then

$$\rho_\alpha(X + Z) < \rho_\alpha(Y + Z).$$

Definition 2 states that decisions over a block, area, phase or cluster of blocks should not depend on the uncertainty of independent elements within the mine. In [14] the authors proved that the only risk measure that complies with additive consistency is the entropic risk measure. Given the uniqueness showed in [14], we will focus our attention in the entropic risk measure and examine its suitability for risk-averse UP problem (4.2.1).

4.3.2 The Entropic risk measure

Let X be a random variable. The entropic risk measure of X at level α is defined as

$$\rho_{\alpha}^{Ent}[X] := \begin{cases} \frac{1}{\alpha} \log \mathbb{E}[e^{\alpha X}] & \text{if } \alpha \neq 0, \\ \mathbb{E}[X] & \text{if } \alpha = 0. \end{cases}$$

If $\alpha > 0$, the decision maker is risk-averse, $\alpha < 0$ means risk-seeker, and $\alpha = 0$ corresponds to the risk-neutral (expected value) case. We will focus on non negative values of α . The entropic risk measure (Ent) is convex, but is not *coherent* [5, 14] as shown in Appendix A.2, Lemma 5.

Our objective is to show that Ent is a viable risk-averse tool for the stochastic UP problem:

Lemma 1. *The entropic risk measure is risk nested when $\rho_{\alpha} = \rho_{\alpha}^{Ent}$ in problem (4.2.2).*

Proof. Proof is available in Appendix A.1.2 □

As a convention, if we are *indifferent* to whether extract a block or not, we will always choose to extract the block to avoid issues that could potentially yield non-nested pits solutions in problem (4.2.2).

In the next section, we will exemplify our approach using different configurations of uncertainty in the ore grade of the blocks of a mine, and compare the pits obtained to the same solution we would obtain by using the commonly used nested pits approach fixing the ore grade to its mean value.

4.4 Computational results

We present two sets of results: first, we consider a small-sized mine which was designed to illustrate the properties and methodologies we propose in this paper. The second mine is a real life instance with a large number of blocks to test the applicability of our methodology in a realistic environment.

We will provide a comparison of the results using different risk measures, and discuss the practical implications and managerial insights of the pits obtained with each method. We implemented all models in Python 3.7.2 and Gurobi 8.1, running on an Intel(R) i7 CPU 7700HQ @ 2.80 GHz and 12 GB of memory over Windows 10.

4.4.1 Small mine

This mine has 36 blocks in a two dimensional configuration, and is defined in terms of its topology and ore grade configuration. Figure 4.3 shows the topology of the mine, and the precedences were configured as follows: if a block is to be extracted, then the blocks on top, right and left of the block on top must be extracted, emulating a 45-degree slope angle.

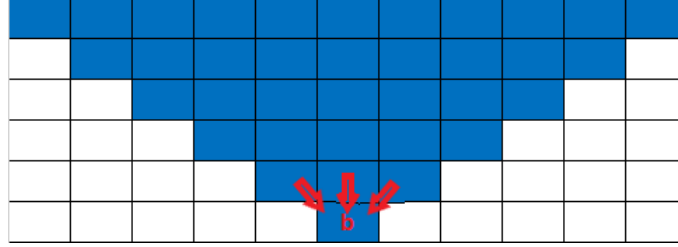


Figure 4.3: Topology of the small mine: blue means the block is part of set of precedent blocks of block b .

One of our goals is to study the behavior of using Ent as a risk measure versus the solutions obtained by using the classical nested pit methodology. In the next Lemma we show that, under normality, we can solve the problem directly, avoiding the use of sampling.

Lemma 2. *If the ore grade of the blocks follows a multivariate normal distribution $\tilde{g} \sim N(\mu, \Sigma)$ where μ is the vector of mean values and Σ is the variance-covariance matrix, then problem (4.2.2) for the entropic risk measure can be formulated as a mixed integer program with a convex quadratic objective function (MIQP)*

Proof. See Appendix A.1.3 for the proof. $\square$

Using Lemma 2, we use a multivariate normal distribution for the ore grades obtaining the following equivalent formulation for UP with Ent:

$$\min_{x^e, x^p \in X^{EP}} \sum_{b \in B} c_b^e x_b^e + \sum_{b \in B} (c_b^p - r_b \bar{g}_b) x_b^p + \frac{1}{2} \alpha \sum_{b \in B} \sum_{b' \in B} r_b x_b^p r_{b'} x_{b'}^p \Sigma_{bb'}, \quad (4.4.6)$$

where X^{EP} is the same feasible set of solutions for problem (4.2.2). Problem (4.4.6) is convex given that matrix Σ is positive semidefinite, and it is amenable to be solved by Gurobi.

Results

We present the results for the following choice of parameters: $c_b^e = 0.5, c_b^p = 0, r_b = 1 \forall b \in B$. Since we have the closed quadratic formulation (4.4.6), we can calculate the exact distribution of the negative profits for each tested model.

The small mine was created as shown in Figure 4.4: the mean negative profit of extracting and processing any of the colored blocks as yellow (1) and red (2) always *pays* its extraction and its precedences. The red blocks have a variance greater than 0 and all possible pairs of red blocks have a positive covariance, while the yellow block has a variance of 0. We included the yellow block because we want to study the behavior of the models where there are blocks with no uncertainty to be extracted. All other blocks (blue) are waste, with grade $g_b = 0$.

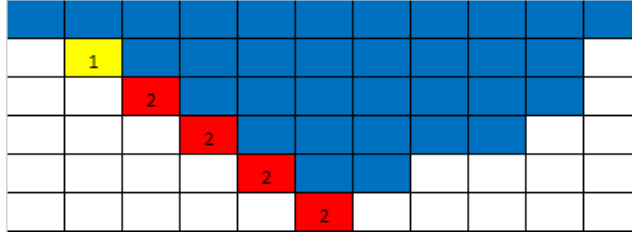


Figure 4.4: Configuration of the blocks in the small mine.

The different pits for Ent are shown in Figure 4.5. It is interesting to note the evolution of the pits when we multiply the parameter Σ by some positive constant: as variability grows, the optimal pit configuration changes for the same values of α . The sub-figures show the evolution of the pits using the following values:

- Green (1): $\alpha = 0$, risk neutral case.
- Yellow (2): $\alpha = 0.1$.
- Orange (3): $\alpha = 0.5$.
- Red (4): $\alpha = 1$.

Note that if any of these values (or colours) are missing, is because the pit obtained is the same as the pit obtained by using a higher value of α , i.e., in Figure 4.5b the optimal solution using 0.5 and 1 are the same.

As uncertainty becomes larger, the pits obtained show a conservative evolution: as α grows the optimal solution pit obtained becomes smaller. In Figure 4.5c we see that for any value of $\alpha > 10$ all pits obtained extract the yellow block. Therefore, if variability in the ore grade is high then Ent will provide pits with small differences between the solutions when changing the value of α . Please note that, while our objective was to observe problem (4.2.2) using Ent under different conditions of uncertainty, we observed that the solutions obtained were nested, even if the covariance of the ore grades between some of the blocks are not 0.

The same analysis can be performed using the current practice in the industry, namely solving problem (4.2.1) and using different values of β (the classical

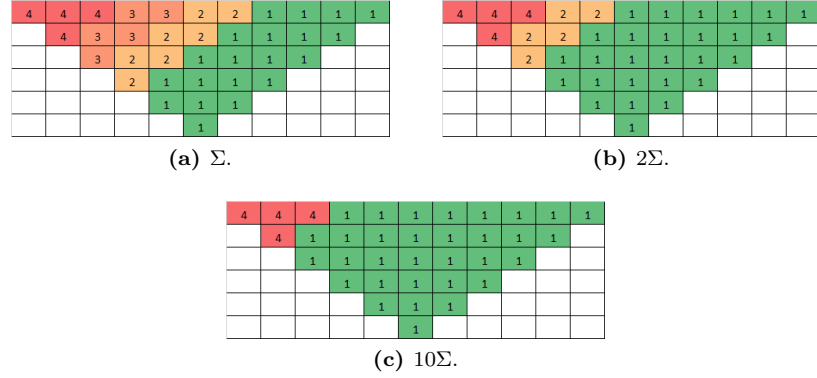


Figure 4.5: Configuration of the pits obtained with different values of α and changing the variance-covariance configuration.

nested pits methodology) to this mine. Note that this methodology doesn't consider the variance of the grades, only its mean values. Since the mean value is constant, we get the same pits even in scenarios of high uncertainty. Figure 4.6 shows this, the configuration of matrix Σ does not change the optimal solutions found by using different values of β as described in Section 4.2.2. The evolution of these pits is obtained using the following values:

- Green (1): $\beta = 0$, risk neutral case.
- Red (2): $\beta = 0.5$.
- No blocks are extracted: $\beta \geq 0.7$.

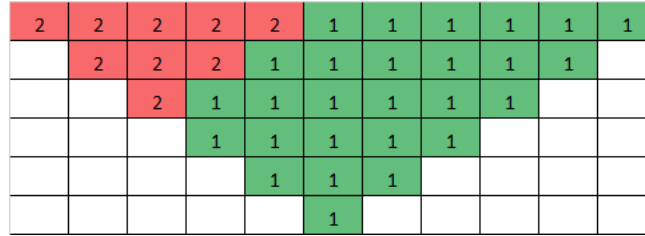


Figure 4.6: Configuration of the pits obtained with different values of β .

The definition of *risk* between models differs: Ent is aiming at controlling variability, and its most conservative result has variance 0. For the nested pit case, since we are manipulating the price, for values of β close to 1 the mining *profit* is small compared to the costs of extraction and processing, which means the corresponding pit will be small, and in the limit will correspond to not extracting any block.

In summary, for a self-constructed small mine the classical approach has the tendency of being overly conservative for some values of β , avoiding extraction

altogether. Moreover, the classical approach is completely insensitive to changes in the correlation between blocks, which is alarming. We can clearly see how variability becomes an important factor with the Ent approach, and how it is controlled by the choice of α : conservative values aim for small but less variable expected profits.

4.4.2 Andina case

The Andina mine is an open pit mine that belongs to Codelco, Chile's state-owned company that controls around 20% of global reserves. Andina is located in Rio Blanco (approximately 80 km NE of Santiago) and is still active after 82 years. A sector of the mine is represented by 26,400 blocks, and the ore distribution is approximated by 120 scenarios generated from a set of drilling holes via conditional simulations using the turning bands algorithm [18].

Since this mine does not have a multivariate normal distribution of ore grades we cannot use the closed quadratic formulation (4.4.6). We tried to solve the problem directly on a smaller version of Andina (less than 100 aggregated blocks) using non-linear optimization solvers (AMPL/MINOS) to check if we could handle the real mine with its 26,400 blocks. The solver was not able to close an optimality gap of 99.9% after 5 hours running, which means it is hopeless to try to attack the problem directly.

Ent is a non-linear function, and since we have integer variables the resulting risk-averse UP problem is extremely challenging to solve for larger mines. However, given the convexity of the exponential function, we can use a piecewise linear approximation of the objective function. In order to accomplish this, we modify the objective function of (4.2.2) thanks to the translation invariance of Ent (please refer to Lemma 6) and the monotonicity of the log function. All necessary proofs and final model which approximates the value of problem (4.2.2) using Ent can be found in Appendix A.3.

Results

We divide the set of 120 ore grade scenarios in a subset of 20 samples which is used to solve the problem, and the remaining 100 are used for the out-of-sample analysis.

Our piecewise linear approximation was designed to be an uniform grid between integers -20 and 20 with steps of 0.01 (4,000 steps in total), because Gurobi treats values below 10^{-8} as 0, and $\exp(-20) \approx 2.06 \cdot 10^{-9}$. Therefore, we calculated a scaling factor and we changed the `NumericFocus` parameter of Gurobi to its maximum value in order to give more processing power in the numerical calculations, and to avoid numerical issues given the small numbers that will be used by our model. We use the mean absolute deviation (MAD) to check the performance of our approximation.

We denote by NP the results obtained by the nested pit methodology. In each case, we evaluate the total profit of the resulting ultimate pit on each of the out-of-sample scenario independently. This give us a set of 100 values for each

solution. We also compare these values with the *lower bound* (LB) obtained solving problem (4.2.1) fixing g_b to each scenario. LB will have a different pit and negative profit per scenario, serving as a benchmark of the best possible result. Also, we present the variance coefficient (VC), which is the ratio between standard deviation and average of the profits.

Table 4.2 shows the results for NP, while changing the value of β . If $\beta \geq 0.7$, the solutions obtained have less than 100 blocks extracted and for $\beta \geq 0.9$ it doesn't extract blocks, therefore these results are omitted. This behavior was already shown in the small mine example in the previous section: conservative values of β will incur in avoiding extraction altogether. However, as risk-averseness grows, the VC becomes larger (6.9% in average).

<i>Model</i>	LB	NP 0	NP 0.1	NP 0.2	NP 0.3	NP 0.4	NP 0.5	NP 0.6
<i>Upit blocks</i>	-	15,775	12,738	7,303	4,218	1,963	816	310
<i>Average</i>	-132.1	-121.4	-110.7	-86.5	-64.5	-43.2	-24.6	-12.4
<i>VC</i>	-	5.9%	5.6%	5.3%	5.6%	6.3%	8.1%	11.0%
<i>% vs LB</i>	-	91.9%	83.8%	65.5%	48.8%	32.7%	18.6%	9.4%

Table 4.2: Table of descriptive statistics for NP of the out of sample results.

Table 4.3 shows the results for Ent. We observe a larger amount of blocks extracted in the pit of the neutral case, and using larger values of α the pits obtained have less blocks. Please note, as risk-averseness grows, the value of the VC becomes smaller (5.6% in average). The MAD for different values of α are of order $O(10^{-5})$ in the worst case, showing that our linear approximation is effective. Values for $\alpha > 12.5$ are not present in the table since the model incurs in numerical errors: the solution obtained was an empty pit.

<i>Model</i>	LB	Ent 0	Ent 7.5	Ent 10	Ent 12.5
<i>Upit blocks</i>	-	21,770	19,545	14,810	11,273
<i>MAD</i>	-	-	2.62E-07	2.55E-07	2.92E-07
<i>Average</i>	-132.1	-128.8	-125.6	-110.7	-98.6
<i>VC</i>	-	5.9%	5.7%	5.3%	5.4%
<i>% vs LB</i>	-	97.5%	95.1%	83.8%	74.6%

Table 4.3: Table of descriptive statistics for Ent of the out of sample results.

Figure 4.7 shows the negative profit obtained per scenario of Ent, NP and LB on each of the testing set ore grade samples, sorted by the results of LB and using different value of α and β respectively.

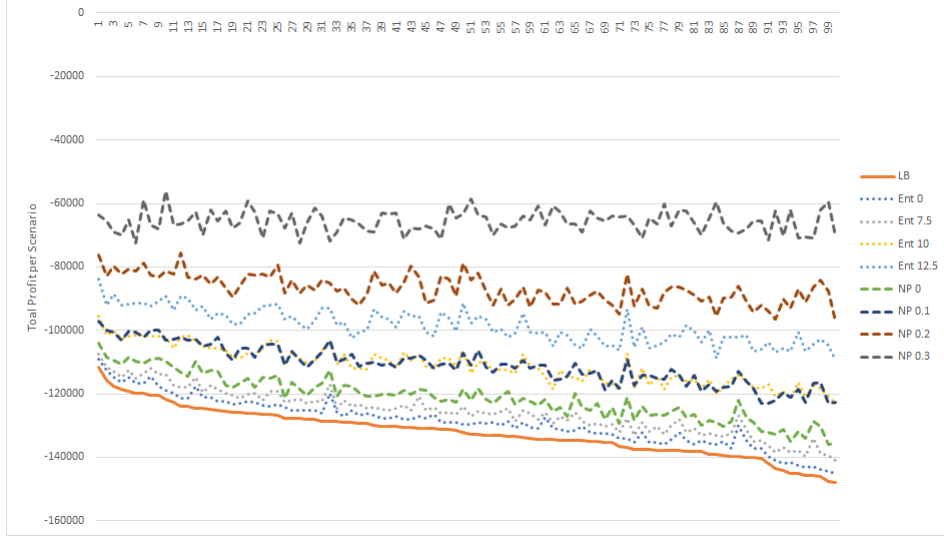


Figure 4.7: Negative profits per scenario for the Andina case.

Figure 4.8 shows a cross sectional view of the pit, where we can observe the differences on the optimal solutions of Ent and NP. We used $\alpha = 10$ and $\beta = 0.03$ to obtain pits with a similar number of blocks extracted (14,810 in Ent versus 14,983 in NP.). We can see both models aim for different sections of the mine.

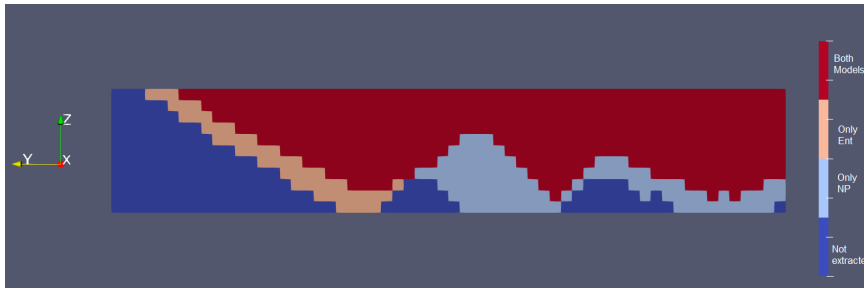


Figure 4.8: Comparing pits of both models.

The importance of these results are the effective application of an approximation method to prove Ent as a viable tool for risk-aversion in real life instances of mining optimization problems. The error obtained in the approximation to the real value of the exponential function is of order $O(10^{-5})$, which was within our acceptance tolerance and Gurobi was able to solve the approximation formulation in an average time of under 5 minutes for all the values of α .

4.5 Conclusions

In this paper, we formulate and discuss how to solve the risk-averse UP problem. We start by describing two properties that a risk measure should satisfy in this context: *risk nestedness* and *additive consistency*. Risk nestedness is a fundamental property since modern mine scheduling algorithms use the idea of nested pits as an input for generating a sequence of block extractions over time. Additive consistency is a desirable property that is intuitive in the sense that it preserves preferences in the presence of independent waste blocks. An interesting consequence of our work is that one of the most popular risk measures, the Conditional Value-at-Risk, fails to satisfy both properties even in the case of blocks with independent distribution, calling into question its use in mining.

In a small simulated mine we contrast results for the current methodology of changing sale prices to obtain nested pits, the classical nested pit (NP) approach, and the proposed Entropic risk measure methodology (Ent). The former behaves as expected: we obtain different pits that vary from mining everything up to avoid working on the mine altogether in the most conservative cases. The Ent approach shows a different behavior: we obtain a different pit transition as we vary the level of risk-aversion, more focused on avoiding variance within the results, thus high uncertainty scenarios will have smaller pits than low uncertainty cases for the same value of α .

We apply our method to a real-life mine (Andina, in Chile) and use a linear approximation to efficiently solve the risk averse problem with the entropic risk measure. Our results show the approximation errors were within tolerable margins, validating the linear approximation scheme. Future work includes extensions to our numerical algorithm to be able to cope with larger mines with millions of blocks, and the study of dynamic risk-averse models for mine scheduling problems.

Chapter 5

Bibliography

- [1] Fouad Ben Abdelaziz, Belaid Aouni, and Rimeh El Fayedh. Multi-objective stochastic programming for portfolio selection. *European Journal of Operational Research*, 177(3):1811–1823, 2007.
- [2] Ahmad Abdi and Ricardo Fukasawa. On the mixing set with a knapsack constraint. *Mathematical Programming*, 157(1):191–217, 2016.
- [3] Shabbir Ahmed and Dimitri J Papageorgiou. Probabilistic set covering with correlations. *Operations Research*, 61(2):438–452, 2013.
- [4] Antonio Alonso-Ayuso, Felipe Carvallo, Laureano F Escudero, Monique Guignard, Jiaying Pi, Raghav Puranmalka, and Andrés Weintraub. Medium range optimization of copper extraction planning under uncertainty in future copper prices. *European Journal of Operational Research*, 233(3):711–726, 2014.
- [5] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent Measures of Risk. *Mathematical Finance*, 9(3):203–228, 1999.
- [6] John E Beasley. An algorithm for set covering problem. *European Journal of Operational Research*, 31(1):85–93, 1987.
- [7] Patrizia Beraldi and Andrzej Ruszczyński. The probabilistic set-covering problem. *Operations Research*, 50(6):956–967, 2002.
- [8] S. Ahmed B.K. Pagnoncelli and A. Shapiro. Sample Average Approximation Method for Chance Constrained Programming: Theory and Applications. *Journal of Optimization Theory and Applications*, 142(2):399–416, mar 2009.
- [9] Louis Caccetta and Stephen P. Hill. An application of branch and cut to open pit mine scheduling. 2003.

- [10] Marco C Campi and Simone Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008.
- [11] Marco C Campi, Simone Garatti, and Maria Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149–157, 2009.
- [12] Abraham Charnes, William W Cooper, and Gifford H Symonds. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4(3):235–263, 1958.
- [13] John W Chinneck. *Feasibility and Infeasibility in Optimization:: Algorithms and Computational Methods*, volume 118. Springer Science & Business Media, 2007.
- [14] Roberto Cominetti and Alfredo Torrico. Additive Consistency of Risk Measures and Its Application to Risk-Averse Routing in Networks. *Mathematics of Operations Research*, 41(4):1510–1521, 2016.
- [15] Tanisha G Cotton and Lewis Ntaimo. Computational study of decomposition algorithms for mean-risk stochastic linear programs. *Mathematical Programming Computation*, 7(4):471–499, 2015.
- [16] Noel Cressie. The origins of kriging. *Mathematical Geology*, 1990.
- [17] Darinka Dentcheva, András Prékopa, and Andrzej Ruszczyński. Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming*, 89(1):55–77, 2000.
- [18] Xavier Emery and Christian Lantuéjoul. Tbsim: A computer program for conditional simulation of three-dimensional gaussian random fields via the turning bands method. *Computers & Geosciences*, 32(10):1615–1628, 2006.
- [19] D Espinoza, G Lagos, E Moreno, and J P Vielma. Risk Averse Approaches In Open-Pit Production Planning under ore Grade uncertainty: a Ultimate Pit Study. 2013.
- [20] J Gleeson and J Ryan. Identifying minimally infeasible subsystems of inequalities. *ORSA Journal on Computing*, 2(1):61–63, 1990.
- [21] John Gleeson and Jennifer Ryan. Identifying minimally infeasible subsystems of inequalities. *ORSA Journal on Computing*, 2(1):61–63, 1990.
- [22] J Golamnejad, M Osanloo, and B Karimi. A chance-constrained programming approach for open pit long-term production scheduling in stochastic environments. *Journal of the Southern African Institute of Mining and Metallurgy*, 106(2):105–114, 2006.

- [23] Marcos Goycoolea, Daniel G. Espinoza, Eduardo Moreno, and Orlando Rivera Letelier. Comparing new and traditional methodologies for production scheduling in open pit mining. *Application of Computers and Operations Research in the Mineral Industry - Proceedings of the 37th International Symposium, APCOM 2015*, pages 352–359, 2015.
- [24] Tito Homem-de Mello and Bernardo K Pagnoncelli. Risk aversion in multistage stochastic programming: A modeling and algorithmic perspective. *European Journal of Operational Research*, 249(1):188–199, 2016.
- [25] Tito Homem-De-Mello and Bernardo K Pagnoncelli. Risk aversion in multistage stochastic programming: A modeling and algorithmic perspective. *European Journal of Operational Research*, 249(1):188–199, 2016.
- [26] Simge Küçükyavuz. On mixing sets arising in chance-constrained programming. *Mathematical programming*, 132(1-2):31–56, 2012.
- [27] Mustafa Kumral. Application of chance-constrained programming based on multi-objective simulated annealing to solve a mineral blending problem. *Engineering Optimization*, 35(6):661–673, 2003.
- [28] Guido Lagos, Daniel Espinoza, Eduardo Moreno, and Jorge Amaya. Robust planning for an open-pit mining problem under ore-grade uncertainty. *Electronic Notes in Discrete Mathematics*, 37(C):15–20, 2011.
- [29] Miguel Lejeune and Nilay Noyan. Mathematical programming approaches for generating p-efficient points. *European Journal of Operational Research*, 207(2):590–600, 2010.
- [30] Miguel A Lejeune. Pattern-based modeling and solution of probabilistically constrained optimization problems. *Operations research*, 60(6):1356–1372, 2012.
- [31] Helmut Lerchs and Grossman FI. Optimum design of open-pit mines. In *Operations Research*, volume 12, page B59. Inst Operations Research Management Sciences, 1964.
- [32] James Luedtke and Shabbir Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699, 2008.
- [33] Denis Marcotte and Josiane Caron. Ultimate open pit stochastic optimization. *Computers and Geosciences*, 51:238–246, 2013.
- [34] E Moreno, X Emery, M Goycoolea, N Morales, and G Nelis. A two-stage stochastic model for open pit mine planning under geological uncertainty. 2017.
- [35] Włodzimierz Ogryczak and Andrzej Ruszczyński. From stochastic dominance to mean-risk models: semideviations as risk measures. *European Journal of Operational Research*, 116(1):33–50, 1999.

- [36] B.K. Pagnoncelli, Shabbir Ahmed, and A Shapiro. Sample average approximation method for chance constrained programming: theory and applications. *Journal of optimization theory and applications*, 142(2):399–416, 2009.
- [37] G. Parra. Generación de benchmark de fondos para el sistema de pensiones en chile: Un enfoque basado en optimización estocástica. Master’s thesis, Universidad de Chile, 2016.
- [38] András Prékopa. Dual method for the solution of a one-stage stochastic programming problem with random rhs obeying a discrete probability distribution. *Zeitschrift für Operations Research*, 34(6):441–461, 1990.
- [39] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41, 1997.
- [40] Anureet Saxena, Vineet Goyal, and Miguel A Lejeune. Mip reformulations of the probabilistic set covering problem. *Mathematical programming*, 121(1):1–31, 2010.
- [41] A. Shapiro, D. Dentcheva, and A. Ruszczyński. Lectures on stochastic programming: modeling and theory. *Technology*, page 447, 2009.
- [42] Yongjia Song, James R Luedtke, and Simge Küçükyavuz. Chance-constrained binary packing problems. *INFORMS Journal on Computing*, 26(4):735–747, 2014.
- [43] Matthew W Tanner and Lewis Ntaimo. IIS branch-and-cut for joint chance-constrained stochastic programs and application to optimal vaccine allocation. *European Journal of Operational Research*, 207(1):290–296, 2010.
- [44] Matthew W. Tanner and Lewis Ntaimo. IIS branch-and-cut for joint chance-constrained stochastic programs and application to optimal vaccine allocation. *European Journal of Operational Research*, 207(1):290–296, nov 2010.
- [45] Minkang Zhu, Daniel B Taylor, Subhash C Sarin, Randall Kramer, and Others. Chance constrained programming models for risk-based economic and policy analysis of soil conservation. *Agricultural and Resource Economics Review*, 23(1):58–65, 1994.

Appendix A

Risk-averse UP Problem

A.1 Mathematical proofs

A.1.1 Risk-nestedness for the UP problem of the entropic measure

Lemma 3. *Let $\alpha_1, \alpha_2 \in \mathbb{R}$, with $\alpha_1 > \alpha_2$. For any random variable X we have that $\rho_{\alpha_1}^{Ent}(X) \geq \rho_{\alpha_2}^{Ent}(X)$.*

Proof. We will show that the objective function of (4.2.2) increases monotonically with $\alpha > 0$ (since $\alpha = 0$ is the expected value case and $\alpha < 0$ is the risk seeking case). Let us define $f(X, \alpha)$ as:

$$f(X, \alpha) = \frac{1}{\alpha} \log \mathbb{E}[e^{\alpha X}],$$

where X is a random variable. Now, let us study the behavior of the first partial derivative over α :

$$\frac{\partial}{\partial \alpha} f(X, \alpha) = \frac{1}{\alpha^2} \left(\frac{\mathbb{E}[\alpha X e^{\alpha X}]}{\mathbb{E}[e^{\alpha X}]} - \log \mathbb{E}[e^{\alpha X}] \right).$$

Let $Y = e^{\alpha X}$, then:

$$\begin{aligned} \frac{\partial}{\partial \alpha} f(X, \alpha) &= \frac{1}{\alpha^2} \left(\frac{\mathbb{E}[Y \log Y]}{\mathbb{E}[Y]} - \log \mathbb{E}[Y] \right) \\ &= \frac{1}{\alpha^2} \left(\frac{\mathbb{E}[Y \log Y] - \mathbb{E}[Y] \cdot \log \mathbb{E}[Y]}{\mathbb{E}[Y]} \right). \end{aligned}$$

Since $\frac{1}{\alpha^2} > 0$ and $\mathbb{E}[Y] = \mathbb{E}[e^{\alpha X}] > 0$, we just need to study the sign of the numerator in the partial derivative. Let $h(Y) = Y \log Y$, then:

$$\mathbb{E}[Y \log Y] - \mathbb{E}[Y] \log \mathbb{E}[Y] = \mathbb{E}[h(Y)] - h(\mathbb{E}[Y]).$$

Finally, since $h(Y)$ is a convex function ($\frac{\partial^2 h(Y)}{\partial Y^2} = \frac{1}{Y} > 0$) then by using Jensen's inequality we know that $\mathbb{E}[h(Y)] - h(\mathbb{E}[Y]) \geq 0$, therefore $f(X, \alpha)$ increases monotonically. $\square$

Lemma 3 ensures the monotonic behavior of Ent w.r.t. the value of α .

Lemma 4. *Let X, Y are two independent random variables and $0 < \alpha < \infty$, then*

$$\rho_\alpha^{Ent}(X + Y) = \rho_\alpha^{Ent}(X) + \rho_\alpha^{Ent}(Y)$$

Proof. Please refer to [14] for the proof. $\square$

A.1.2 Proof of Lemma 1

Proof. Let $\tilde{g}_b$ be the random variable of the ore grade for each block $b \in B$ and assume they are independently distributed, and let $\alpha_1, \alpha_2 \in \mathbb{R}$ where $0 < \alpha_2 < \alpha_1 < \infty$. Suppose that a certain block b is in the pit obtained by solving (4.2.2) using Ent with α_1 , but is not present in the pit under the same problem solved by using α_2 . Let U_1 be the pit obtained in the first case and U_2 the second case.

Since extracting 0 blocks is a feasible solution and $\rho_\alpha^{Ent}(0) = 0$ for $\alpha > 0$ then we know that $\rho_{\alpha_1}^{Ent}(U_1) \leq 0$. Furthermore, $\rho_{\alpha_1}^{Ent}(U_1 \setminus U_2) \leq 0$, else $U_1 \cap U_2$ would obtain a better value for α_1 but U_1 is the optimal solution which would be a contradiction.

By using Lemma 3, since $\alpha_2 < \alpha_1$, then $\rho_{\alpha_2}^{Ent}(U_1 \setminus U_2) \leq \rho_{\alpha_1}^{Ent}(U_1 \setminus U_2) \leq 0$. But then the pit $U_2 \cup (U_1 \setminus U_2)$ is also feasible and by Lemma 4 we have that $\rho_{\alpha_2}^{Ent}(U_2 \cup (U_1 \setminus U_2)) = \rho_{\alpha_2}^{Ent}(U_2) + \rho_{\alpha_2}^{Ent}(U_1 \setminus U_2) \leq \rho_{\alpha_2}^{Ent}(U_2)$, which is a contradiction of the optimality condition of U_2 , finishing the proof. $\square$

A.1.3 Proof of Lemma 2

Proof. Let $\tilde{g} \sim N(\mu, \Sigma)$ then let Z be as follows:

$$Z := \left(\sum_{b \in B} c_b^e x_b^e + \sum_{b \in B} (c_b^p - r_b \tilde{g}_b) x_b^p \right) \rightarrow N(\mu', \sigma'^2),$$

where

$$\mu' = \sum_{b \in B} c_b^e x_b^e + \sum_{b \in B} (c_b^p - r_b \bar{g}_b) x_b^p,$$

and

$$\begin{aligned} \sigma'^2 &= (r \circ x^p)^T \Sigma (r \circ x^p) \\ &= \sum_{b \in B} \sum_{b' \in B} r_b x_b^p r_{b'} x_{b'}^p \Sigma_{bb'}, \end{aligned}$$

where operator $\circ$ is the Hadamard (element-wise) matrix product.

We can verify that $\mathbb{E}[e^{\alpha Z}]$ is the moment generating function of a univariate normal distribution, which in turn corresponds to:

$$\exp\left(\alpha\mu' + \frac{1}{2}\alpha^2\sigma'^2\right).$$

Therefore, we obtain the following equivalent formulation:

$$\min_{x^e, x^p \in X^{EP}} \rho^{Ent} \left(\sum_{b \in B} c_b^e x_b^e + \sum_{b \in B} (c_b^p - r_b \tilde{g}_b) x_b^p \right) = \min_{x^e, x^p \in X^{EP}} \mu' + \frac{1}{2}\alpha\sigma'^2,$$

where X^{EP} is the same feasible set of solutions for problem (4.2.2), and since Σ is s.d.p., the proof is finished. $\square$

A.2 Other Lemmas and Propositions of interest

Lemma 5. *The entropic risk measure is not a coherent risk measure.*

Proof. Let X be a random variable and $a \geq 0$. We have:

$$\rho_\alpha^{Ent}(aX) = \frac{1}{\alpha} \log \mathbb{E}[e^{\alpha a X}].$$

If $\beta = \alpha a$, then

$$\begin{aligned} \rho_\alpha^{Ent}(aX) &= \frac{a}{\beta} \log \mathbb{E}[e^{\beta X}] \\ &= a \rho_\beta^{Ent}(\beta X) \neq a \rho_\alpha^{Ent}(X). \end{aligned}$$

Therefore, Ent is not positive homogeneous, finishing the proof. $\square$

Lemma 6. *The entropic risk measure is a translation invariant risk measure.*

Proof. Let X be a random variable and $a \geq 0$. We have:

$$\begin{aligned} \rho_\alpha^{Ent}(X + a) &= \frac{1}{\alpha} \log \mathbb{E}[e^{\alpha(X+a)}] \\ &= \frac{1}{\alpha} \log \mathbb{E}[e^{\alpha X} e^{\alpha a}] \\ &= \frac{1}{\alpha} \log e^{\alpha a} \mathbb{E}[e^{\alpha X}] \\ &= \frac{1}{\alpha} (\log e^{\alpha a} + \log \mathbb{E}[e^{\alpha X}]) \\ &= \frac{1}{\alpha} (\alpha a + \log \mathbb{E}[e^{\alpha X}]) \\ &= \frac{1}{\alpha} \log \mathbb{E}[e^{\alpha X}] + a \\ &= \rho_\alpha^{Ent}(X) + a. \end{aligned}$$

Therefore, Ent is translation invariant, finishing the proof. $\square$

Proposition 5. *If $\rho(\cdot) = \mathbb{E}[\cdot]$, then formulation (4.2.1) is equivalent to (4.2.2).*

Proof. It can be seen that in formulation (4.2.2) we are using two sets of variables: x^e set of blocks to be extracted and x^p set of blocks to be processed. In (4.2.3) we have the constraint $x_b^p \leq x_b^e \forall b \in B$, which limits the possibility to process a block only if it is extracted in the first place, then for a certain block b the cost in the objective function can be one of three possible values:

1. 0 if the block is not extracted, i.e. $x_b^e = 0$.
2. c_b^e if the block is extracted but not processed, i.e. $x_b^e = 1$ and $x_b^p = 0$.
3. $c_b^p - r_b g_b^\omega + c_b^e$ if the block is extracted and processed, i.e. $x_b^e = 1$ and $x_b^p = 1$.

As (4.2.3)'s objective is to minimize the negative profit of processing the blocks given $\tilde{g}$, then the decision can be stated as:

$$x_b^p = \begin{cases} 1, & \text{if } c_b^p < r_b g_b^\omega, \\ 0, & \text{if } c_b^p \geq r_b g_b^\omega. \end{cases}$$

On the other hand, the expression $c_b^e - \mathbb{E}[(r_b \tilde{g}_b^\omega - c_b^p)^+]$ can assume exactly the same three previously described options. Since we can pre-compute the value of $\mathbb{E}[(c_b^p - r_b g_b^\omega)^+]$, $\forall b \in B$, we can eliminate the use of a second set of variables. If $x_b = 1$ in (4.2.1) it means that the block must be extracted whether is waste or is going to be processed. $\square$

A.3 Entropic risk objective function linearization

The risk-averse UP problem using Ent can be written as

$$\min_{x^e, x^p, \omega \in X^{EP\Omega}} \frac{1}{N} \sum_{\omega \in \Omega} \exp \left(\alpha \left(\sum_{b \in B} c_b^e x_b^e + \sum_{b \in B} (c_b^p - r_b g_b^\omega) x_b^{p, \omega} \right) \right), \quad (\text{A.3.1})$$

where Ω is the domain of the random variable g_b and

$$X^{EP\Omega} := \{x^e \in \{0, 1\}^{|B|}, x_b^{p, \omega} \in \{0, 1\}^{|B \times \Omega|} : x_{b'}^e \leq x_b^e \forall (b, b') \in P, \\ x_b^{p, \omega} \leq x_b^e \forall b \in B, \forall \omega \in \Omega\}.$$

We will use two auxiliary variables: v_ω and z_ω , where

$$\sum_{\omega \in \Omega} v_\omega \geq \frac{1}{N} \sum_{\omega \in \Omega} \exp(z_\omega).$$

The strategy here is to use an approximation of the exponential function for each term in the sum of the objective function in (A.3.1), using the auxiliary variable v_ω . Problem (A.3.1) is equivalent to the following problem:

$$\begin{aligned} \text{UP}_e &:= \min_{x^e, x^p \in X^{EP\Omega}} \sum_{\omega \in \Omega} v_\omega \\ \text{s.t.} \quad & z_\omega \geq \alpha \left(\sum_{b \in B} c_b^e x_b^e + \sum_{b \in B} (c_b^p - r_b g_b^\omega) x_b^{p, \omega} \right) \quad \forall \omega \in \Omega \quad (\text{A.3.2}) \\ & N v_\omega \geq \exp(z_\omega) \quad \forall \omega \in \Omega, \\ & v_\omega \geq 0 \quad \forall \omega \in \Omega. \end{aligned}$$

The non-linear constraint of (A.3.2) will be replaced by a piecewise linear approximation, detailed in the work of [37], for each constraint represented by z_ω . For differentiable convex functions (such as $f(x) = \exp(x)$), we can use a lower approximation using the gradient at a given point t_i : $f(x) - f(t_i) \geq \nabla f(t_i)(x - t_i)$. We will select a set of K points $U := \{t_1, \dots, t_K\}$ to calculate the value of $\nabla f(t_i) \quad \forall i \in U$ and approximate the exponential function using linear functions as shown in Figure A.1.

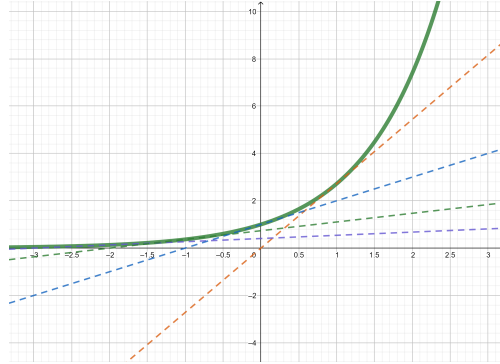


Figure A.1: Subgradient linear approximation of $\exp(x)$ using $K = 4$ support points where $U = \{-2, -1, 0, 1\}$.

Using this in (A.3.2), we get the following constraints:

$$v_\omega \geq \exp(u_\omega - t_i) + \exp(t_i) \quad \forall \omega, i \in \Omega \times U,$$

which in turn can be used to construct our approximated problem:

$$\begin{aligned}
\text{UP}_A &:= \min_{x^e, x^p \in X^{EP\Omega}} \sum_{\omega \in \Omega} v_\omega \\
\text{s.t.} \quad & z_\omega \geq \alpha \left(\sum_{b \in B} c_b^e x_b^e + \sum_{b \in B} (c_b^p - r_b g_b^\omega) x_b^{p,\omega} \right) \quad \forall \omega \in \Omega \text{ (A.3.3)} \\
& v_\omega - e^{t_i} z_\omega \geq (1 - t_i) e^{t_i} \quad \forall \omega, i \in \Omega \times U, \\
& v_\omega \geq 0 \quad \forall \omega \in \Omega.
\end{aligned}$$