UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE CIENCIAS

Escuela Profesional de Matemáticas

SEMINARIO DE TESIS II

Decomposition algorithms applied to multistage stochastic linear programs

Elaborado por
David Willians Morante Moran
Asesores
Eladio Ocaña Anaya
Ernesto Oré Albornoz

LIMA-PERÚ
2021

Contents

Contents **i**

1 Introduction **1**
1.1 Motivation 1
1.2 Work organization 2

2 Theoretical Concepts **5**
2.1 Notation 5
2.2 Stochastic Linear Programming 6
2.2.1 Two-Stage Program with recourse 6
2.2.2 Multistage Linear Program with recourse 11
2.3 Monotone Operator Theory 18
2.3.1 Definitions 18
2.3.2 Resolvent Operator and Proximal Point Algorithm . . . 20
2.3.3 Relation with Convex Analysis 23

3 Nested decomposition **25**
3.1 L-shaped Algorithm 25
3.1.1 Feasiblity cuts 26
3.1.2 Optimality cuts 27
3.2 Multicut L-shaped Algorithm 30
3.3 Nested L-shaped Algorithm 32
3.3.1 Convergence Analysis 36

4 Progressive-Hedging Algorithm **39**
4.1 Scenario Aggregation Principle 39
4.1.1 Projected Problem 43
4.2 A view of PH from Monotone Operators 44

4.3 Douglas-Rachford splitting 44
4.4 Spingarn partial inverse 47
4.5 Convergence Analysis 48

5 Applications **53**
5.1 Multistage Problem 56
5.2 Two Stage Problem 58
5.2.1 Progressive Hedging 59
5.2.2 Parameter r 62
5.2.3 L-shaped Method 62

6 Conclusions **65**

Bibliography **67**

Chapter 1

Introduction

1.1 Motivation

In real life, we often encounter problems in which we are forced to make a decision without knowing what future effects it will have. Problems with these characteristics are treated within the subfield of mathematics called Stochastic Optimization. One of these problems is the Hydrothermal Dispatch problem, which consists of minimizing the total cost of hydro and thermal generation by satisfying user demand over a planning horizon. This task is one of the most challenging in the field of optimization and for many years the problem was modeled with a deterministic approach [11],[12]. However, this way was not the most adequate due to many factors, one of them is the storage of water resources, which depends a lot on the information of the incoming flows that we have. This hydrological information is of a random nature, and therefore we must consider its randomness in the model, otherwise, we could incur in an underestimation or overestimation of operating costs. There are many more factors in conditions of uncertainty, such as user demand, for example, in summer, demand tends to increase because it is more common to use refrigerants and air conditioners and this makes it even more difficult to manage the operation of the electrical system. [15] shows some first results of considering hydrologic uncertainty in the dispatch problem. In such studies stochastic programming helped to evaluate the impact of an operational decision by formulating various scenarios. But with the gain of making a decision considering risks, the computational cost required to solve the problem was also increased. This is because the size of the problem grows as more scenarios are evaluated. Thus, motivated by finding a balance between the accuracy of a decision and the time that the computational calculation would take, new methods were developed whose main idea is the decomposition of the global problem in small problems that are easier and faster to solve, which, in addition, can be solved in parallel.

1.2 Work organization

In this introductory work, we will study two solution methods, the Nested L-shaped Algorithm and the Progressive Hedging Algorithm, which exploit the decomposable structure of a certain class of optimization problems (in particular linear stochastic problems with recourse). Below, we explain in a little more detail how these methods will be approached.

Chapter 2 : Theoretical Concepts

In this chapter we introduce the notation that will be used throughout this work along with the necessary theoretical background on stochastic programming and monotone operator theory which will be used to give a different approach to the Progressive Hedging Algorithm.

Capítulo 3: Nested Decomposition

In this chapter we will start by showing the two-stage linear stochastic problem in the nested formulation and two algorithms to solve it, which are the L-shaped Algorithm and a variation of it called the Multicut L-shaped Algorithm. Finally we will talk about the nested formulation of the multistage linear stochastic problem and solve it with the Nested L-shaped Algorithm, which is an extension of the L-shaped Algorithm for more than two stages.

Capítulo 4: Progressive Heging

This chapter shows the explicit formulation of the nonanticipativity constraint of a multistage linear stochastic problem, for this purpose , the concept of "scenario packages" will be introduced. Under this concept a technique developed by Rockafeller and Wets [2] called Progressive Hedging Algorithm will be shown. Finally, the theory of monotonous operators explained in Chapter 2 will be used to give a different approach to this technique.

Capítulo 5: Aplicaciones

This chapter will show numerical results of the presented methods applied to a uninodal problem considering only water generation. For this problem we will compare the solutions obtained by the L-shaped and the progressive hedging algorithms as well the convergence times. In addition, the choice of the parameter r involved in the progressive hedging algorithm will be discussed.

Chapter 2

Theoretical Concepts

This chapter provides a review and introduction to pertinent concepts that will be used throughout this work. First we will introduce the notation to be used. The second section discusses concepts of stochastic linear programming such as the two-stage, multi-stage problem and their comparisons. The third part of this chapter will be devoted to maximal monotone operators, the resolvent operator and proximal point methods with some applications such as the Douglas-Rachford splitting method and the Spingarn partial inverse.

2.1 Notation

We use $\mathbb{R}$ to refer to the real numbers and $\overline{\mathbb{R}}$ to denote the set of extended reals, that is, $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$. $\mathbb{R}^n$ indicates the space of all real vector with n coordinates and $\mathbb{R}^{m \times n}$ the space of real matrices with m rows and n columns.

Denote by $(\Omega, \mathcal{F}, P)$ a probability space where Ω is the sample space of some random experiment, $\mathcal{F}$ a sigma algebra of subsets of Ω and $P : \mathcal{F} \to [0, 1]$ a probability measure. Let $\xi : \Omega \to \mathbb{R}^N$ a random variable which assigns numerical characteristics to the experiment results, then we denote by $\mathbb{E}_\xi$ the expectation of the random variable ξ.

Let $\Omega_1, \ldots, \Omega_T$ sample spaces. For all $t = 1, \ldots, T$, we will use ω_t to represent a particular result of a experiment with sample space Ω_t. Similarly, the random variable that characterizes Ω_t will be donted with $\xi_t : \Omega_t \to \mathbb{R}^{N_t}$.

2.2 Stochastic Linear Programming

A stochatic linear program (SLP) is a special kind of linear programming problem in which a part or all of coefficients are uncertain elements, this can often be modeled with random variables ξ to which the theory of probability can be applied (as will be shown later). But the values that can take the random variables ξ can only be considered once the experiment has taken place ($\omega \in \Omega$). Therefore, it is necessary to introduce the concept of "stages", which indicate the before and after the presence of the uncertain information. If this information is revealed sequentially then there will be multiple stages. This is often confused with the concept of "period", which represents a change in time which is not necessarily associated with the disclosure of the uncertain information.

2.2.1 Two-Stage Program with recourse

Stochastic linear problems with only two stages are called Two-stages linear programs. Schematically the process can be illustrated as:

$$\begin{array}{ccccc} \underline{\text{First Stage}} & & & & \underline{\text{Second Stage}} \\ \text{first decision } x & \rightarrow & \text{observation } \omega & \rightarrow & \text{second decision } y \end{array}$$

The first decision x is made based on some initial information before observing ω from a random experiment. A random variable ξ will be in charge of parameterizing such observation, then a second decision y will be taked based on the above information. In this section we will study problems where the objective function is linear as well as the restrictions and our task will be to find optimal x and y decisions.

Consider some probability space $(\Omega, \mathcal{F}, P)$. Let $\xi(\omega) : \Omega \rightarrow \mathbb{R}^N$ be some random variable on Ω. A two-stage linear stochastic program with recourse has the following mathematical representation:

$$\begin{array}{rl} \min & c^\top x + \mathbb{E}_\xi[\min q(\omega)^\top y(\omega)] \\ s.t & Ax = b \\ & M(\omega)x + W(\omega)y(\omega) = h(\omega) \\ & x \geq 0, y(\omega) \geq 0. \end{array} \tag{$\mathcal{P}$}$$

where

- The vector $x \in \mathbb{R}^{n_1}$ is called first-stage decision (also called recourse action), with associated first-stage cost $c \in \mathbb{R}^{n_1}$ and matrices $b \in \mathbb{R}^{m_1}$, $A \in \mathbb{R}^{m_1 \times n_1}$.

- The random data is given by the random variable which we represent with abuse of notation as

$$\xi(\omega) = (q(\omega), h(\omega), M(\omega), W(\omega))$$

- The vector $y(\omega) \in \mathbb{R}^{n_2}$ is called second-stage decision (also called recourse decision). It is taken after the experiment was observed.

We will associate problem $(\mathcal{P})$ with a deterministic problem, for this, we define the **second-stage value function**

$$Q(x, \xi(\omega)) = \min_y \{q(\omega)^\top y : W(\omega)y = h(\omega) - M(\omega)x, y \geq 0\} \tag{2.1}$$

which is the optimal value in the second-stage after havin taken the first-stage decision x and observing the realization ω. Let the **expected second-stage value function**

$$\mathcal{V}(x) = \mathbb{E}_\xi[Q(x, \xi(\omega))] \tag{2.2}$$

For some first-stage decision x, the function $Q(x, \xi(\omega))$ could be unbounded below or even non-existent and $\mathcal{V}(x)$ as well. Thus, we are interested in a set of first-stage decisions for which $\mathcal{V}(x)$ exists, we deal with that property required for $\mathcal{V}$ introducing the **elementary feasiblity set** defined by

$$\mathcal{X}_2(\xi(\omega)) = \{x \in \mathbb{R}^{n_1} : \exists y \in \mathbb{R}^{n_2}, W(\omega)y = h(\omega) - M(\omega)x, y \geq 0\} \tag{2.3}$$

Since we are seeking that for any situation one can make a recourse decision, it is necessary that the first state decision belongs to the elementary feasible set for any scenario, therefore, we define the **second-stage feasibility set** as:

$$\mathcal{X}_2 = \bigcap_{\omega \in \Omega} \mathcal{X}_2(\xi(\omega)) \tag{2.4}$$

Finally, the problem $(\mathcal{P})$ can be rewritten as

$$\begin{aligned} \min_x \quad & c^\top x + \mathcal{V}(x) \\ s.t \quad & x \in \mathcal{X}_1 \cap \mathcal{X}_2. \end{aligned} \tag{2.5}$$

where $\mathcal{X}_1 = \{x \in \mathbb{R}^{n_1} : Ax = b, x \geq 0\}$ is the set determined by the fixed constraints (not random).

Another representation of the problem $(\mathcal{P})$ can be obtained under the assumption that the set Ω is finite. In this case the function $\mathcal{V}(x)$ is explicitly shown on the model. This formulation is called **deterministic equivalent form** of two-stage problem $(\mathcal{P})$

$$\begin{array}{rll}\min\limits_{x,y(\omega^r),1\leq r\leq R} & c^\top x+\sum_{r=1}^R p_r q(\omega^r)^\top y(\omega^r) & \\ s.t & Ax=b & \\ & M(\omega^r)x+Wy(\omega^r)=h(\omega^r), & r=1,\dots,R \\ & x\geq 0, y(\omega^r)\geq 0, & r=1,\dots,R\end{array} \tag{2.6}$$

where $\Omega=\{\omega^r; r=1,\dots,R\}$ with probabilities $p_r=P(\omega^r), r=1,\dots,R$.

For a small number of scenarios, it is viable to solve problem (2.6) by conventional methods. However, for a large number of scenarios are required to use decomposition techniques.
Note that the constraints have the following structure.

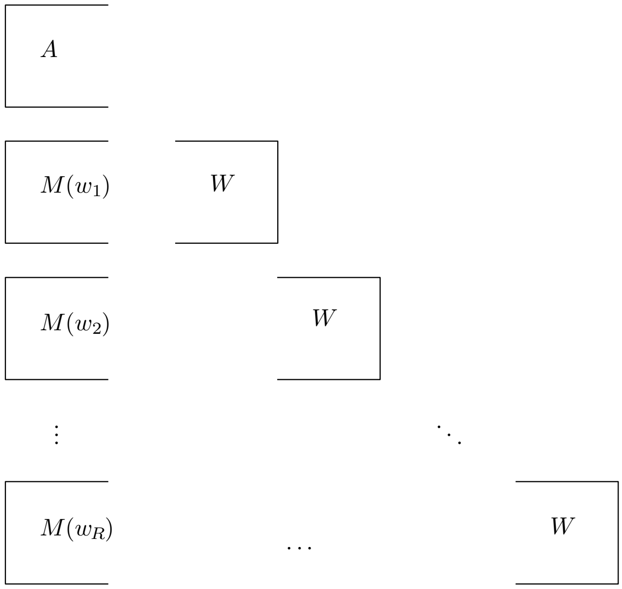

Figure 2.1: Structure of the constraints of the two-stage stochastic problem with fixed recourse in extensive form called L-shaped structure.

The following result allows us to develop efficient decomposition algorithms that will be enunciated in chapter 3 and 4.

Theorem 2.1 *For a finite set Ω and $\xi:\Omega\rightarrow\mathbb{R}^N$ is a discrete random variable, then*

1. *The elementary feasibility set $\mathcal{X}_2(\xi(\omega))$ is a convex polyhedron.*

2. *The second-stage feasibility set $\mathcal{X}_2$ is a convex polyhedron.*

3. *The expected value function $\mathcal{V}(x)$ is a piecewise linear and convex function for all $x \in \mathcal{X}_2$.*

Proof For a matrix A, we define pos$A = \{t : Ay = t, y \geq 0\}$

1. Consider some $x \in \mathbb{R}^{n1}$ such that no $y \geq 0$ exists such that $W(\omega)y = h(\omega) - T(\omega)x$. This is equivalent to say $h(\omega) - T(\omega)x \notin$ pos$W(\omega)$. Thus, we have a point, $h(\omega) - T(\omega)x$, which does not belong to a convex set, pos$W(\omega)$. Then, there must exist some hyperplane, say $\{x : \sigma^\top x = 0\}$, that separates $h(\omega) - T(\omega)x$ from pos$W(\omega)$. This hyperplane satisfies $\sigma^\top t < 0$ for $t \in$ pos$W(\omega)$ and $\sigma^\top(h(\omega) - T(\omega)x) > 0$. Since ξ is fixed, $W(\omega)$ is fixed and there can be only finitely many different such hyperplanes which completes the proof.

2. The intersection of finitely many convex polyhedron is a convex polyhedron.

3. Let $x \in \mathcal{X}_2$, and π the dual simplex multiplier associated to the constraint of $Q(x, \xi(\omega))$. By strong duality

$$Q(x, \xi(\omega)) = \max_{\pi}\{\pi^\top(h(\omega) - M(\omega)x) : W(\omega)\pi \leq q(\omega)\} \quad (2.7)$$

Let $\Lambda = \{\pi : W(\omega)\pi \leq q(\omega)\}$. We denote the extrem points and extrem directions of this pholyedron as π_j and d_i respectively, where $j \in J$ and $i \in I$, with J and I index sets. By the fundamental theorem of linear programming we have

$$Q(x, \xi(\omega)) = \max_{\pi_j}\{\pi^\top(h(\omega) - M(\omega)x)\}, \quad (2.8)$$

$$d_i^\top(h(\omega) - M(\omega)x) \leq 0, \forall i \in I. \quad (2.9)$$

□

from expression (2.8) , we see that $Q(x, \xi(\omega))$ can be expressed as the maximum of linear functions evaluated in x. Hence, is a piecewise linear function in x. To show the convexity of Q just apply the definition. Finally as $\mathcal{V}$ is a convex combination of piecewise linear convex functions then it is also a piecewise linear convex functions for all $x \in \mathcal{X}_2$.

Example 2.2 *Let*

$$\begin{array}{rl} \min & \mathbb{E}_\xi(y_1 + y_2) \\ s.t. & 0 \leq x \leq 10 \\ & y_1 - y_2 = h(\omega) - x \\ & y_1, y_2 \geq 0 \end{array} \quad (2.10)$$

where $h(\omega)$ takes the values 1,2 and 4 with probability 1/3 each one. We observe that

$$Q(x,1) = \min\{y_1 + y_2 : y_1 - y_2 = 1 - x, y_1, y_2 \geq 0\} = \begin{cases} 1 - x & x \leq 1 \\ x - 1 & x \geq 1 \end{cases}$$

$$Q(x,2) = \min\{y_1 + y_2 : y_1 - y_2 = 2 - x, y_1, y_2 \geq 0\} = \begin{cases} 2 - x & x \leq 2 \\ x - 2 & x \geq 2 \end{cases}$$

$$Q(x,4) = \min\{y_1 + y_2 : y_1 - y_2 = 4 - x, y_1, y_2 \geq 0\} = \begin{cases} 4 - x & x \leq 4 \\ x - 4 & x \geq 4 \end{cases}$$

and

$$\mathcal{V}(x) = \begin{cases} 7/3 - x & x \in [0,1] \\ 5/3 - x & x \in [1,2] \\ x/3 + 1/3 & x \in [2,4] \\ x - 7/3 & x \in [4,10] \end{cases}$$

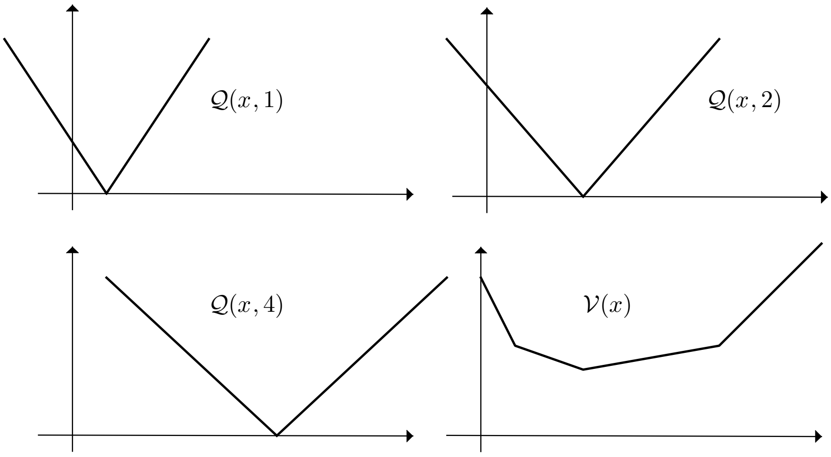

Figure 2.2: Second stage value functions and expected second-stage value function

Special cases of two stochastic programs with recourse

1. When the recourse matrix W is fixed, that is, it is the same for any scenario, then it will be said that the problem $(\mathcal{P})$ is a two-stage stochastic problem with **fixed recourse**. Under this assumption, the following result holds.

   Theorem 2.3 *For a fixed recourse matrix W and $\xi : \Omega \to \mathbb{R}^N$ a random variable with finite second moment ($\mathbb{E}_\xi(\xi^2) < \infty$) , the elemetary* *feasible set can be written as $\mathcal{X}_2(\omega) = \{x \in \mathbb{R}^{n_1} : Q(x, \xi(\omega)) < \infty\}$ for all $x \in \mathcal{X}_2$. [1]*

2. Some stochastic programs have the property that every feasible first-stage decision is also second-stage feasible, that is, $\mathcal{X}_1 \subseteq \mathcal{X}_2$. A stochastic programming $(\mathcal{P})$ with this characteristic is called two-stage stochastic programming with **relatively complete recourse**. Thus, (2.5) is written as:

$$\begin{array}{ll} \min\limits_{x} & c^\top x + \mathcal{V}(x) \\ s.t & Ax = b, x \geq 0. \end{array} \tag{2.11}$$

This representation of a stochastic program with complete recourse shows a computational advantage if we know the exact value of $\mathcal{V}(x)$.

2.2.2 Multistage Linear Program with recourse

In many stochastic programming problem the uncertain data is revealed sequentially causing there to be more than two stages. Such problems are called multistage stochastic programs (MLP) with recourse, and consist of an initial decision followed by several stages of increasing information and further decision making. Schematically the process can be illustrated as:

$$\begin{array}{c} \text{first decision } x_0 \\ \downarrow \\ \text{observation } \in \Omega_1 \\ \downarrow \\ \text{second decision } x_1 \\ \downarrow \\ \vdots \\ \downarrow \\ \text{observation } \in \Omega_{T-1} \\ \downarrow \\ \text{T-th decision } x_{T-1} \end{array}$$

In this scheme we can see the progressive arrival of new information and the increasing of stages, furthermore, it is observed that the decisions at each stage depend only on the information revealed up to that moment and not on future observations, generalizing the two-stage program with recourse shown in the previous section.

If the sample spaces $\Omega_1, \ldots, \Omega_{T-1}$ are finite, a better representation of the whole process is reflected in a decision tree with an initial node (representing the decision of the first stage) followed by roots (representing the uncertainty).The increase in information as time progresses can be seen in the number of terminal leaves of the tree. We define a scenario as a path from the node to one of the leaves, therefore, the total number of scenarios is the total number of leaves.

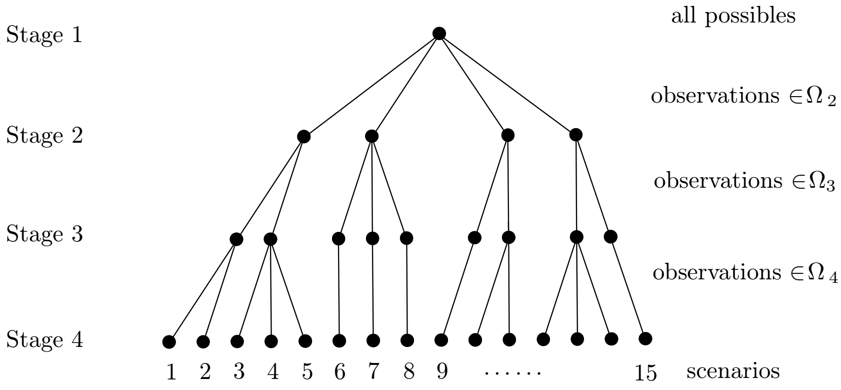

Figure 2.3: Decision tree representation of a four stage stochastic program.

There are many ways to formulate a multistage stochastic problem with recourse. In this work we will present three forms; nested formulation, dynamic formulation and nonanticipative formulation.

A. Nested Formulation

For a multistage linear stochastic problem with recourse, the **nested formulation** is given by

$$\begin{aligned}&\min_{x_0\in F_0}\left[c_0^\top x_0 + \mathbb{E}_{\xi_{[1,T]}|\omega_{[0,0]}}\left[\min_{x_1\in F(x_0,\xi_1(\omega_1))}\left[c_1^\top x_1 + \right.\right.\right.\\ &\qquad \left.\left.\left. + \mathbb{E}\left[\cdots + \mathbb{E}_{\xi_{[T-1,T-1]}|\omega_{[0,T-2]}}\left[\min_{x_{T-1}\in F(x_{T-2},\xi_{T-1}(\omega_{T-1}))} c_{T-1}^\top x_{T-1}\right]\cdots\right]\right.\right.\right.\end{aligned} \tag{2.12}$$

where

- $F_0 = \{x \in \mathbb{R}^{n_0} : Ax = b, x \geq 0\}$ and $c_0 \in \mathbb{R}^{n_0}, b \in \mathbb{R}^{m_0}, A \in \mathbb{R}^{m_0\times n_0}$ is the deterministic first-stage data.
- $F(x_{t-1}, \xi_t(\omega_t)) = \{x \in \mathbb{R}^{n_t} : M_t(\omega_t)x_{t-1} + W_t(\omega_t)x = h_t(\omega_t), x \geq 0\}$ is the feasible region of the stage t problem which depends on the decision in stage $t-1$ and the information $\xi_t(\omega_t)$ available in stage t.

- $\xi_{[t,t']}$ denote the sequence of random data vectors corresponding to stages t through t'. and $\omega_{[t,t']}$ a particular realization (this can be seen as a path from the node in stage t to the t'). For instance $\omega_{[0,0]} = \omega_0$ is the deterministic data in the first-stage, just one realization.
- $\mathbb{E}_{\xi_{[t,T-1]}|\omega_{[0,t-1]}}$ denotes the conditional expectation operator in stage t and gives us the expected value with respect to future observations given that a historical data is fixed.

B. Dynamic Formulation

Let's look at the final stage problem

$$\min_{x_{T-1}} \{c_{T-1}^\top x_{T-1} : M(\omega_{T-1})x_{T-2} + W(\omega_{T-1})x_{T-1} = h(\omega_{T-1}), x_{T-1} \geq 0\} \tag{2.13}$$

this optimal value depends on the decision vector x_{T-2} and data given by ω_T. At the stage $T-2$ we know x_{T-3} and $\omega_{[0,T-2]}$. Therefore, in stage $T-2$, we face with the following stochastic programming problem

$$\begin{array}{ll} \min\limits_{x_{T-2}} & c_{T-2}^\top x_{T-2} + \mathbb{E}_{\xi_T|\omega_{[0,T-2]}}[Q_{T-1}(x_{T-2}, \xi_{T-1})] \\ s.t. & M(\omega_{T-2})x_{T-3} + W(\omega_{T-2})x_{T-2} = h(\omega_{T-2}) \\ & x_{T-2} \geq 0 \end{array} \tag{2.14}$$

The optimal value of the above problem is denoted by $Q_{T-1}(x_{T-2}, \xi_{[0,T-1]})$, in general at the stage $t = 1, \dots, T-2$ we have the problem

$$\begin{array}{rll} Q_t(x_{t-1}, \xi_{[0,t]}) = & \min\limits_{x_t} & c_t^\top x_t + \mathbb{E}_{\xi_{[t+1,T]}|\omega_{[0,t]}}[Q_{t+1}(x_t, \xi_{[0,t+1]})] \\ & s.t. & M(\omega_t)x_{t-1} + W(\omega_t)x_t = h(\omega_t) \\ & & x_t \geq 0 \end{array} \tag{2.15}$$

The optimal value $Q_t(x_{t-1}, \xi_{[0,t]})$ is called **cost-to-go** function. When $t = 0$ we have the problem

$$\begin{array}{ll} \min\limits_{x_0} & c_0^\top x_0 + \mathbb{E}_{\xi_{[1,T]}|\omega_0}[Q_2(x_1, \xi_{[0,2]})] \\ s.t. & Ax_0 = b \\ & x_0 \geq 0 \end{array} \tag{2.16}$$

which does not depend on ω_0 because it is deterministic. Therefore, (2.16) is equivalent to a two-stage stochastic problem. The dynamic programming equations here take the form

$$Q_t(x_{t-1}, \xi_{[0,t]}) = \min_{x_t \in F(x_{t-1}, \xi_t(\omega_t))} \{c_t^\top x_t + V_{t+1}(x_t, \xi_{[0,t]})\} \tag{2.17}$$

where $\mathcal{V}_{t+1}(x_t, \xi_{[0,t]}) = \mathbb{E}_{\xi_{[t+1,T]}|\omega_{[0,t]}}[Q_{t+1}(x_t, \xi_{[1,t+1]})]$ and is called the **expected cos-to-go function.**

C. Nonanticipativity Formulation

Let us consider again the two-stage problem in formulation (2.6)

$$\begin{array}{rll} \min\limits_{x,y(\omega^r),1\leq r\leq R} & c^\top x + \sum_{r=1}^R p_r q(\omega^r)^\top y(\omega^r) & \\ s.t & Ax = b & \\ & M(\omega^r)x + W(\omega^r)y(\omega^r) = h(\omega^r), & r = 1,\ldots,R \\ & x \geq 0, y(\omega^r) \geq 0, & r = 1,\ldots,R \end{array}$$

For this problem we are looking for a vector $(x, y(\omega^1), \ldots, y(\omega^R))$ which minimizes the objective function. It is clearly noted that the vector x does not depend on the scenario (the observation ω), in other words, in each scenario the vector x is the same. Thus, we can rewrite Problem (2.6) as follows

$$\begin{array}{rll} \min\limits_{x(\omega^r),y(\omega^r),1\leq r\leq R} & \sum_{r=1}^R p_r[c^\top x(\omega^r) + q(\omega^r)^\top y(\omega^r)] & \\ s.t & Ax(\omega^r) = b & \\ & M(\omega^r)x + W(\omega^r)y(\omega^r) = h(\omega^r), & r = 1,\ldots,R \\ & x(\omega^r) \geq 0, y(\omega^r) \geq 0, & r = 1,\ldots,R \\ & x(\omega^r) = x(\omega^q) & r,q = 1,\ldots,R. \end{array} \tag{2.18}$$

The difference now is that we are not looking for a single vector, but a vector for each scenario. We can say then that we are looking for a function

$$\begin{array}{rccl} X: & \Omega & \rightarrow & \mathbb{R}^n \\ & s & \mapsto & X(s) = (x_0(s), x_1(s)) \end{array} \tag{2.19}$$

such that it satisfies the conditions of (2.18), where we use the change of notation, $s = \omega$, $x_0(s) = x(s)$ and $x_1(s) = y(s)$.

There are two constraints that we want to emphasize. First, note that

$$X(s) \in C(s), \forall s \in S \tag{2.20}$$

where

$$C(s) = \{(x_0, x_1) \in \mathbb{R}^n : Ax_0 = b, M(s)x_0 + W(s)x_1 = h(s), x_0 \geq 0, x_1 \geq 0\}.$$

Second, note that

$$x_0(s^r) = x_0(s^q), \forall s, q = 1,\ldots,R. \tag{2.21}$$

This constraint (2.21) is called “nonanticipativity” , and it tells us that the first stage decision does not depend on the scenario. If we ignore this constraint, then we would be solving a problem that is separable by scenarios:

$$\begin{array}{rll} \min\limits_{x_0(\omega^r),x_1(\omega^r),1\le r\le R} & \sum_{r=1}^{R} p_r[c^\top x_0(\omega^r)+q(\omega^r)^\top x_1(\omega^r)] & \\ s.t & Ax_0(\omega^r)=b & \\ & M(\omega^r)x_0+W(\omega^r)x_1(\omega^r)=h(\omega^r), & r=1,\dots,R \\ & x_0(\omega^r)\ge 0, x_1(\omega^r)\ge 0, & r=1,\dots,R. \end{array} \tag{2.22}$$

We now want to extend the formulation for a multistage problem. First, we should note that for a two-stage problem, the set of scenarios is equivalent to the sample space, however in a multi-stage problem the set of scenarios is not necessarily the cartesian product $\Omega_1 \times \cdots \times \Omega_T$, but rather the set of paths from the node to the leaves.

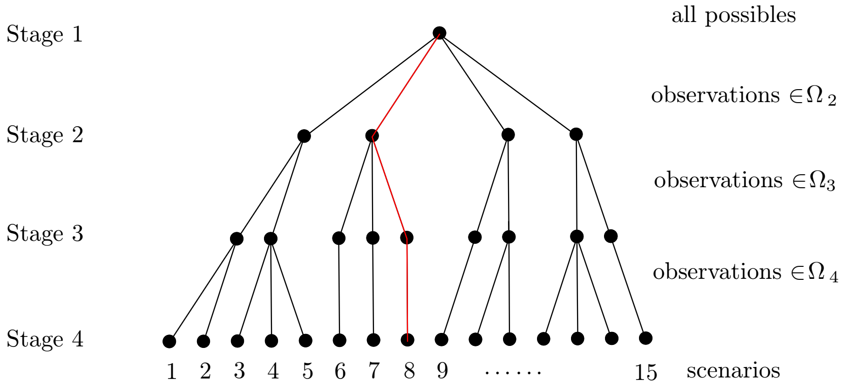
Stage 1
Stage 2
Stage 3
Stage 4
all possibles
observations $\in\Omega_2$
observations $\in\Omega_3$
observations $\in\Omega_4$
1 2 3 4 5 6 7 8 9 15
scenarios

Figure 2.4: Particular scenario. The red line is the scenario 8.

Furthermore, since we are looking for a vector for each scenario (obtaining a function) instead of a single vector, then the tree of scenarios is described in a different way called the nonanticipative scenario tree, see Figure 2.5.

Let a T-stages stochastic program with the set of scenarios definded as

$$S=\{s^1,s^2,\dots,s^R\}.$$

Any scenario $s_r \in S$ has $T-1$ parts (see for example Figure 2.5 with T=4), which correspond to the uncertain information revealed between the stages. So, we can write the scenario s^r as:

$$s^r=(s_1^r,s_2^r,\dots,s_t^r,\dots,s_{T-1}^r). \tag{2.23}$$

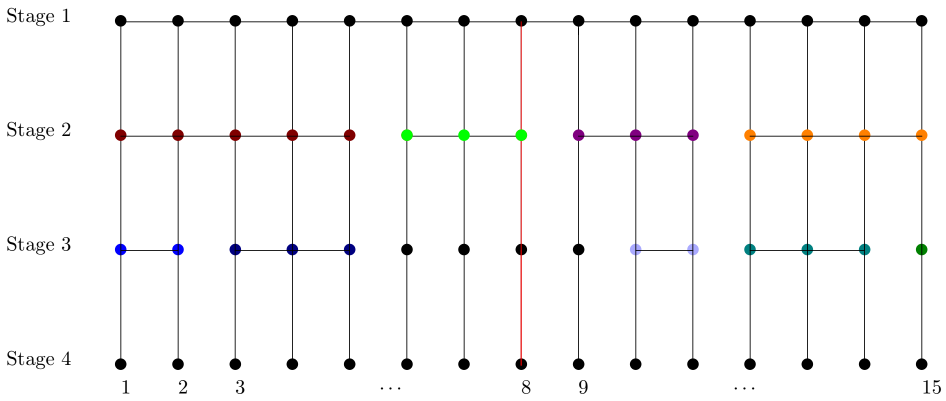

Figure 2.5: Nonanticipative scenario tree. The horizontal lines indicates the same node in the scenario tree of Figure 2.4. The red line is the scenario 8.

Then a T-stage stochastic linear programming with explicit nonanticipativity constraint is:

$$\begin{array}{lll} \min & \sum_{r=1}^{R} p_r[c_0^\top x_0(s^r) + c_1^\top x_1(s^r) + \cdots + c_{T-1}^\top x_{T-1}(s^r)] & \\ s.t & Ax_0(s^r) & = b \\ & M_1(s^r)x_0(s^r) + W_1(s^r)x_1(s^r) & = h_1(s^r) \\ & \qquad\qquad \ddots & = \vdots \\ & \qquad\qquad M_{T-1}(s^r)x_{T-2}(s^r) + W_{T-1}(s^r)x_{T-1}(s^r) & = h_{T-1}(s^r) \\ & x_t(s^r) \geq 0, \forall t = 0, \dots, T-1; \forall r = 1, \dots, R & \\ & x_t(s^r) = x_t(s^q), \forall s^r, s^q \in S \text{ such that } s^r_{[0,t]} = s^q_{[0,t]} & \end{array} \tag{2.24}$$

where p_r is the probability that scenario s^r occurs, which is calculated as the multiplication of the observations probabilities that were given form that scenario, that is, s_t^r for all t. We use the notation

$$s^r_{[0,t]} := (s^r_1, s^r_2, \dots, s^r_t)$$

hence, $s^r_{[0,t]} = s^q_{[0,t]}$ indicates that the revealed data up to stage t for scenario s^r and stage s^q must be the same.

We can write (2.24) in a more compact form. For that, let's define the called **policy set**:

$$\mathcal{E} = \{X : S \to \mathbb{R}^n : X(s) = (x_0(s), x_1(s), \dots, x_{T-1}(s)) \in \mathbb{R}^{n_0} \times \dots \times \mathbb{R}^{n_{T-1}}\} \tag{2.25}$$

A policy X will be said to be **admissible**, if for each scenario $s \in S$ we have that $X(s) \in C(s)$. The set of all **admissible policies** will be denoted by $\mathcal{C}$, then

$$\mathcal{C} = \{X \in \mathcal{E} : X(s) \in C(s), \forall s \in S\}$$

In terms of policies, a map $X : S \to \mathbb{R}^n$ is called nonanticipative if for two scenarios s^r and s^q that are indistinguishable from the start up the time t (i.e. $s^r_{[0,t]} = s^q_{[0,t]}$), holds $X_t(s^r) = X_t(s^q)$. A good way to model nonanticipativity is introducing the concept of "scenario bundles":

At each stage t, form a partition $\mathcal{A}_t$ of S whose elements are disjoint subsets of scenarios indistinguishable up to stage t called **scenario bundles**. That is X is a nonanticipative policy if and only if for all t and any scenario bundle $A \in \mathcal{A}_t$ we have

$$s^r, s^q \in A \Rightarrow X_t(s^r) = X_t(s^q)$$

Then, we define the set of nonanticipativity policies as

$$\mathcal{N}_S = \{X \in \mathcal{E} : X_t \text{ is constant on each scen. bundle } A \in \mathcal{A}_t \text{ of } S, \forall t\}$$

From now on we will refer to a nonanticipative policy, as an **implementable policy**. It is not difficult to notice that $\mathcal{N}_S$ is a linear subspace of $\mathcal{E}$

Notation 2.4 *Since we already know that the set of implementable policies depends on the set of scenarios, we will dispense with this subscript and use only $\mathcal{N}$ instead of $\mathcal{N}_S$*

Finally, the compact form of (2.24) is as follows:

$$\begin{array}{rl} \min\limits_{X \in \mathcal{E}} & \sum\limits_{s \in S} p(s) f_s(X(s)) \\ s.t & X \in \mathcal{C} \cap \mathcal{N} \end{array} \tag{2.26}$$

where $f_s(X(s)) = c_0^\top x_0(s) + c_1^\top x_1(s) + \cdots + c_{T-1}^\top x_{T-1}(s)$. If we obviate the nonanticipativity constraint, then the problem would be separable into the following scenario subproblems:

$$\begin{array}{rl} \min\limits_{X \in \mathcal{E}} & p(s) f_s(X(s)) \\ s.t & X \in \mathcal{C}(s) \end{array} \tag{$\mathcal{P}_s$}$$

2.3 Monotone Operator Theory

The theory of set-valued monotone operators plays a central role in many areas of nonlinear analysis. In convex analysis, for example, we will see that the subdifferential operator is a monotone maximal operator. The main concept that will be shown in this section is the proximal operator, which will be useful for the study of the progressive-hedging algorithm that solves the problem (2.24).

2.3.1 Definitions

Let $T : \mathbb{R}^n \to \mathbb{R}^n$ be an operator (a map that acts on elements of a space to produce elements of another space, possibly the same space), a point x is said to be a **fixed point** of T if $x = Tx$. The set of all fixed points of T is denoted by $\operatorname{Fix} T$ (this set can be empty)

$$\operatorname{Fix} T = \{x \in \mathbb{R}^n : x = Tx\}$$

The operator $T : \mathbb{R}^n \to \mathbb{R}^n$ is said to be **firmly nonexpansive** if

$$\forall x \in \mathbb{R}^n, \forall y \in \mathbb{R}^n, \|Tx - Ty\|^2 + \|(\operatorname{Id} - T)x - (\operatorname{Id} - T)y\|^2 \leq \|x - y\|^2 \tag{2.27}$$

where $\operatorname{Id} : \mathbb{R}^n \to \mathbb{R}^n$ is the identity operator. The operator $T : \mathbb{R}^n \to \mathbb{R}^n$ is said to be **β-Lipshitz continuous** with constant $\beta \in \mathbb{R}_{++}$ if

$$\forall x \in \mathbb{R}^n, \forall y \in \mathbb{R}^n, \|Tx - Ty\| \leq \beta \|x - y\| \tag{2.28}$$

When $\beta = 1$, T is said to be **nonexpansive operator**. If $\beta < 1$ then T is a **contraction mapping**. If T is firmly nonexpansive, clearly is nonexpansive, but the reciprocal is not true (consider $T = -\operatorname{Id}$).

Nonexpansive operators play an important role in this work. We will see applications that are reduced to finding a fixed point for this type of operator. An example of this is the following theorem known as Karnosel'skii-Mann algorithm.

Theorem 2.5 *Let D be a nonempty closed convex subset of $\mathbb{R}^n$, let $T : D \to D$ be a nonexpansive operator such that $\operatorname{Fix} T \neq \emptyset$, let $(\lambda_n)n \in \mathbb{N}$ be a sequence in $[0, 1]$ such that $\sum n\lambda_n(1 - \lambda_n) = +\infty$, and let $x_0 \in D$. Set*

$$x_{n+1} = \lambda_n T x_n + (1 - \lambda_n)x_n. \qquad \forall n \in \mathbb{N} \tag{2.29}$$

Then the sequence x_n converges to a point in $\operatorname{Fix} T$.

A **multivalued operator** in $\mathbb{R}^n$ is denoted by $A : \mathbb{R}^n \to 2^{\mathbb{R}^n}$and means that A maps every point $x \in \mathbb{R}^n$ to a set $Ax \subseteq \mathbb{R}^n$. This operator $A : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ is usually characterized by its graph

$$\operatorname{gra} A = \{(x, u) \in \mathbb{R}^n \times \mathbb{R}^n : u \in Ax\}$$

The domain and range of multivalued operator A are the sets

$$\operatorname{dom} A = \{x \in \mathbb{R}^n : Ax \neq \emptyset\}, \qquad \operatorname{ran} A = A(\mathbb{R}^n) = \bigcup_{x \in \mathbb{R}^n} Ax \tag{2.30}$$

A one-valued operator $T : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ can be corresponded with the multivalued operator defined as

$$A : \mathbb{R}^n \to 2^{\mathbb{R}^n} : x \mapsto \begin{cases} \{Tx\}, & x \in D \\ \emptyset, & x \notin D \end{cases}$$

and vice versa. The **inverse** of A, denoted by A^{-1} is defined through its graph

$$\operatorname{gra} A^{-1} = \{(u, x) \in \mathbb{R}^n \times \mathbb{R}^n : (x, u) \in \operatorname{gra} A\}$$

where, for all $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n, u \in Ax \Leftrightarrow x \in A^{-1}u$. The **set of zeros** of $A : \mathbb{R}^n \to 2^{\mathbb{R}^n}$, denoted by $\operatorname{zer} A$ is the set

$$\operatorname{zer} A = A^{-1}0 = \{x \in \mathbb{R}^n : 0 \in Ax\} \tag{2.31}$$

Many applications in nonlinear analysis can be reduced to finding an element of $\operatorname{zer} A$ when $A : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ is a **monotonous operator**, i.e.,

$$\forall (x, u) \in \operatorname{gra} A, \forall (y, v) \in \operatorname{gra} A, \quad \langle x - y, u - v \rangle \geq 0 \tag{2.32}$$

also when A is a **maximal monotone operator**,which means that there is no monotonous operator $B : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ such that $\operatorname{gra} B$ is a proper subset of $\operatorname{gra} A$.

The **partial inverse** of $A : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ with respect to a linear closed subspace V of $\mathbb{R}^n$ is the operator $A_V : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ defined by its graph

$$\operatorname{gra} A_V = \{(P_V x + P_{V^\perp} u, P_V u + P_{V^\perp} x) : (x, u) \in \operatorname{gra} A\}. \tag{2.33}$$

where $P_V x$ represents the orthogonal projection of x onto V. We have that $A_{\mathbb{R}^n} = A$ and $A_{\{0\}} = A^{-1}$, so the partial inverse can be seen as an object between the operator and its inverse. Also it is not difficult verify that A_V is monotone (maximal) if and only if A is.

Given the operators $A : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ and $B : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ y $\lambda \in \mathbb{R}_{++}$, is defined the operator $A + \lambda B$ as

$$A + \lambda B : \mathbb{R}^n \to 2^{\mathbb{R}^n} : x \to Ax + \lambda Bx \tag{2.34}$$

Then $\operatorname{gra}(A + \lambda B) = \{(x, u + \lambda v) : (x, u) \in \operatorname{gra} A, (x, v) \in \operatorname{gra} B\}$ and $\operatorname{dom}(A + \lambda B) = \operatorname{dom} A \cap \operatorname{dom} B$.

2.3.2 Resolvent Operator and Proximal Point Algorithm

The **resolvent** of $A : \mathbb{R}^n \to 2^{\mathbb{R}^n}$, denoted by J_A is

$$J_A = (\operatorname{Id} + A)^{-1} \tag{2.35}$$

with graph given by $\operatorname{gra} J_A = \{(x+u, x) \in \mathbb{R}^n \times \mathbb{R}^n : (x,u) \in \operatorname{gra} A\}$. The following result gives us some properties of the resolvent operator of a maximal monotonous operator

Proposition 2.6 *Let $A : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ be maximally monotone and let $\gamma > 0$. Then the following hold:*

1. *$J_{\gamma A} : \mathbb{R}^n \to \mathbb{R}^n$ and $\operatorname{Id} - J_{\gamma A} : \mathbb{R}^n \to \mathbb{R}^n$ are firmly nonexpansive and maximally monotone.*
2. *The **reflected operator** defined as*

$$R_{\gamma A} : \mathbb{R}^n \to \mathbb{R}^n : x \mapsto 2J_{\gamma A}x - x \tag{2.36}$$

   is nonexpansive.

3. $\operatorname{Fix} J_{\gamma A} = \operatorname{zer} A$
4. *If V is a linear subspace of $\mathbb{R}^n$ then $y = J_{A_V}x \Leftrightarrow P_V y + P_{V^\perp}(x-y) = J_A x$.*

The third property tells us that it is enough to find a fixed point of the operator $J_{\gamma A}$ to find a zero of the operator A.

Theorem 2.7 (Proximal-point algorithm) *Let A be a maximally monotone operator such that $\operatorname{zer} A \neq \emptyset$, let (γ_n) be a positive sequence such that $\sum_{n\in\mathbb{N}} \gamma_n^2 = +\infty$, and let $x_0 \in \mathbb{R}^n$. Set*

$$x_{n+1} = J_{\gamma_n A} x_n \tag{2.37}$$

then x_n converges to a point in $\operatorname{zer} A$.

Let $T : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$, and let $\alpha \in (0,1)$. Then T is called **α-averaged**, if there exists a nonexpansive operator $R : D \to \mathbb{R}^n$ such that

$$T = (1-\alpha)\operatorname{Id} + \alpha R$$

The α-averaged map T is instead a nonexpansive operator, but the reciprocal is not true, consider $T = -\operatorname{Id} : \mathbb{R}^n \to \mathbb{R}^n$.

Example 2.8 *Let A and B be maximal monotone. The operator $T_{\gamma A,\gamma B}$ defined as*

$$T_{\gamma A,\gamma B} = J_{\gamma A}R_{\gamma B} + \mathrm{Id} - J_{\gamma B} \tag{2.38}$$

is $1/2$ averaged. $T_{\gamma A,\gamma B}$ is called Douglas-Rachford operator and was introduced by Lions and Mercier [12].

An operator is $1/2$-averaged if and only if it is firmly non-expansive. Then the Douglas-Rachford operator is firmly nonexpansive.

If T is α-average, and $\lambda_n \in [0, 1/\alpha]$, the Karsnosel'skii-Mann algorithm holds since α-averaged implies nonexpansiveness. If in addition $\alpha = 1/2$ we obtain the following result which is a version of the fixed point theorem for firmly nonexpansive operators.

Corollary 2.9 *Let $T : \mathbb{R}^n \to \mathbb{R}^n$ be a firmly nonexpansive operator such that $\operatorname{Fix} T \neq \emptyset$, let $x_0 \in \mathbb{R}^n$, and set $x_{n+1} = Tx_n$. Then the sequence x_n converges to a point in $\operatorname{Fix} T$.*

Two applications of the proximal fixed point theorem are now shown

A. Spingarn's partial inverse.

The method of partial inverse is an application of the proximal-point algorithm to the resolvent of the partial inverse operator. Specifically, if $A : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ is a maximal monotone operator and V is a closed linear subspace of $\mathbb{R}^n$, the problem

$$\text{find } x \in V \text{ and } u \in V^{\perp} \text{ such that } u \in Ax \tag{2.39}$$

is solved by setting

$$z_{n+1} = J_{A_V} z_n \tag{2.40}$$

where $z_n = x_n + u_n$ with $x_n \in V$ and $u_n \in V^{\perp}$ for all $n \in \mathbb{N}$. Which implies that $z^* = (x^*, u^*) \in A_V$ solve (2.39) with $x_n \to x^* = P_V z^*$ and $u_n \to u^* = P_{V^{\perp}} z^*$. By item 4 of the Proposition 2.7 we can write (2.40) as

$$P_V z_{n+1} + P_{V^{\perp}}(z_n - z_{n+1}) = J_A z_n.$$

Making $y_n = J_A z_n$ and $v_n = x_n + u_n - y_n$ we obtain the following iterations, starting from $x_0 \in V$ and $u_0 \in V^{\perp}$

$$\begin{aligned} y_n &= J_A(x_n + u_n) \\ v_n &= x_n + u_n - y_n \\ (x_{n+1}, u_{n+1}) &= (P_V y_n, P_{V^{\perp}} v_n) \end{aligned}$$

Which is known as the Spingarn's partial inverses method.

B. Douglas Rachford Splitting method.

Let A and B maximally monotone operators in $\mathbb{R}^n$ such that $\operatorname{zer}(A+B) \neq \emptyset$, the problem of

$$\text{find } x \in \mathbb{R}^n \text{ such that } 0 \in Ax + Bx \tag{2.41}$$

is solved applying the proximal point algorithm to the Douglas Ratchford Operator $T = J_{\gamma A}R_{\gamma B} + \operatorname{Id} - J_{\gamma B}$, where $\gamma > 0$, because the sum of two maximal monotonic operators is not necessarily maximal monotonous, we cannot directly apply the proixmal point algorithm to J_{A+B}. So we have

$$y_{n+1} = J_{\gamma A}R_{\gamma B}y_n + y_n - J_{\gamma B}y_n \tag{2.42}$$

where $y_n \to y \in \operatorname{Fix} T = \operatorname{Fix} R_{\gamma A}R_{\gamma B}$ ([3]). Making $x_n = J_{\gamma B}y_n$, and $z_n = R_{\gamma B} = 2x_n - y_n$, we obtain the following iterations

$$\begin{aligned} x_n &= J_{\gamma B}y_n, \\ z_n &= J_{\gamma A}(2x_n - y_n), \\ y_{n+1} &= y_n + (z_n - x_n) \end{aligned}$$

where $x_n \to x^* \in \mathcal{L}$ the set of solution of problem (2.41). Furthemore, if we consider the dual inclusion problem associated to (2.41), i.e.

$$\text{find } u \in \mathbb{R}^n \text{ such that } 0 \in -A^{-1}(-u) + B^{-1}u \tag{2.43}$$

then, by the following caracterization of the set $\mathcal{L}$:

$$\mathcal{L} = \{x \in \mathbb{R}^n : \exists u \in \mathcal{D} \text{ with } -u \in Ax \text{ and } u \in Bx\} \tag{2.44}$$

where $\mathcal{D}$ is the set of solutions of the problem (2.43), and the relation obtained by the Douglas Rachford Algorithm for (2.41)

$$z_n - x_n = J_{\gamma A}(2x_n - y_n) - x_n \tag{2.45}$$

we will have that $(x^*, \gamma^{-1}(y^* - x^*)) \in \operatorname{gra} B$ and $(x^*, -\gamma^{-1}(y^* - x^*)) \in \operatorname{gra} A$. Where we can write

$$u_n = \gamma^{-1}(y_n - x_n) \tag{2.46}$$

and $u_n \to u = \gamma^{-1}(y^* - x^*)$.

2.3.3 Relation with Convex Analysis

As an example for the concepts presented above and at the same time as motivation of what will be developed in the following chapters, we will talk about the subdifferential operator

Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper function.The operator denoted by ∂f and given by

$$\partial f : \mathbb{R}^n \to 2^{\mathbb{R}^n} : x \mapsto \{u \in \mathbb{R}^n : \forall y \in \mathbb{R}^n, \langle y - x, u\rangle + f(x) \leq f(y)\} \tag{2.47}$$

is called subdiferential operator of f. The effective domain of ∂f is

$$\operatorname{dom} \partial f = \{x \in \mathbb{R}^n : \partial f(x) \neq \emptyset\}.$$

The subdifferential operator ∂f is monotonous because for each $u \in \partial f(x)$ and $v \in \partial f(y)$ we have $\langle y - x, u\rangle + f(x) \leq f(y)$ and $\langle x - y, v\rangle + f(y) \leq f(x)$, then adding these inequalities we obtain $\langle x - y, u - v\rangle \geq 0$.

We denote by $\Gamma_0(\mathbb{R}^n)$ to the set of functions $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ proper l.s.c. If $f \in \Gamma_0(\mathbb{R}^n)$ we have the following results

i. ∂f is maximal monotone.[1]

ii. $J_{\partial f}$ is a one value operator given by

$$J_{\partial f} : \mathbb{R}^n \to \mathbb{R}^n : v \mapsto \arg\min_x \left(f(x) + \frac{1}{2}\|x - v\|^2\right)$$

Proof If $z \in (\mathrm{Id} + \partial f)^{-1}(v)$, then $v \in z + \partial f(z)$, equivalent to $0 \in \partial f(z) + (z - v)$. Since $2(z - v) = \nabla\|z - v\|^2$ then $0 \in \partial f(z) + \partial\frac{1}{2}\|z - v\|^2 \Rightarrow 0 \in \partial\left(f(z) + \frac{1}{2}\|z - v\|^2\right)$. Finally $z = \arg\min_x \left(f(x) + \frac{1}{2}\|x - v\|^2\right)$. Take into account that this last equality is due to the fact that the function $f(x) + \frac{1}{2}\|x - v\|^2$ is strictly convex. □

Let $f \in \Gamma_0(\mathbb{R}^n)$. The **proximal operator** of f is defined by

$$\operatorname{prox}_f : \mathbb{R}^n \to \mathbb{R}^n : v \to \arg\min_x \left(f(x) + \frac{1}{2}\|x - v\|^2\right) \tag{2.48}$$

Due to item ii. the resolvent of the subdiferential ∂f of a funciton $f \in \Gamma_0(\mathbb{R}^n)$ is $J_\partial = \operatorname{prox}_f$.

Interpretation of the proximal operator

When we evaluate the proximal f operator at v, we are trying to reduce the value of f, but we are penalized for straying too far from v. If f is the following function

$$f(x) = \begin{cases} 0, & x \in C \\ +\infty, & x \notin C \end{cases} \tag{2.49}$$

where C is a non-empty closed convex set in $\mathbb{R}^n$, the proximal operator of f reduces to the orthogonal projection on C

$$\operatorname{prox}_f(v) = \arg\min_{x \in C} \left(\frac{1}{2} \|x - v\|^2 \right) = P_C v \tag{2.50}$$

Therefore, the proximal operators can be viewed as generalized projections, and this perspective suggests several properties that we expect the proximal operators to obey. Evaluating $prox_f$ at the blue points moves them to the corresponding red points. The three points in the domain of the function stay in the domain and move towards the minimum of the function, while the other two move to the boundary of the domain and towards the minimum of the function.

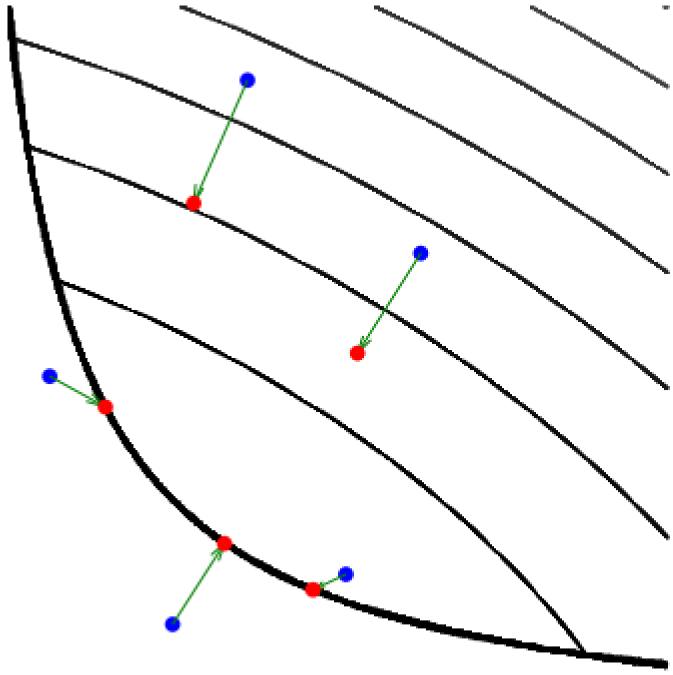

Figure 2.6: Evaluating a proximal operator at various points

Chapter 3

Nested decomposition

The nested decomposition algorithms are efficient cutting plane methods for solving (2.12). These methods exploit the fact that the value function is piecewise linear convex in order to approximate it by hyperplanes iteratively. In this chapter we explain how to obtain this cuts in the L-shaped algorithm, a variation of it, called Multicut L-shaped algorithm and also extend this idea to the resolution of a multistage stochastic problem.

3.1 L-shaped Algorithm

The L-shaped algorithm is is a procedure that allows us to compute the solution of the two-stage stochastic problem (2.6) which could also be expressed in the deterministic equivalent form:

$$\begin{array}{rl} \min\limits_{x} & c^\top x + V(x) \\ s.t & x \in K_1 \cap K_2. \end{array} \tag{3.1}$$

this problem is reformulated as follows

$$\begin{array}{rl} \min\limits_{x} & c^\top x + \theta \\ s.t & V(x) \leq \theta \\ & x \in K_1 \cap K_2. \end{array} \tag{3.2}$$

The method to be used is to decompose the problem (3.2) into a single first-stage problem and a number of second stage problems, one corresponding to each of the possible scenarios that could be realized in stage 2. The first step is to start with an arbitrary first-stage decision, if we try to solve the problem of the second stage with this initial choice, two situations could occur :

1. Some of the subproblems are infeasible or
2. All the subproblems are feasible.

Both of these situations give us a way to add constraints to the problem. In the first situation we will be provided with constraints known as **feasibility cuts** and in the second we will obtain constraints known as **optimality cuts**. Define the master problem

$$\begin{array}{rlll} \min & c^\top x + \theta & & \\ s.t. & Ax = b & & \\ & E_l x + \theta \geq e_l & l = 1, \dots, s & \\ & D_l x \geq d_l & l = 1, \dots, m & \\ & x \geq 0, \theta \in \mathbb{R} & & \end{array} \tag{3.3}$$

where the vaues of D_l, d_l, E_l and e_l will be specified in the two subsections below

3.1.1 Feasiblity cuts

The idea of this step is to determine when a first-stage decision is a feasible second-stage solution. If this is not the case, then the decision, as it is not feasible for the second stage, would be infeasible for the entire problem.

Suppose the first-stage decision is an arbitrary $x \in K_1 = \{Ax = b, x \geq 0\}$. For each possible scenario in stage 2, we have the following linear problem

$$\begin{array}{rl} Q_r(x) = \min & q(\omega^r)^\top y \\ s.t. & M(\omega^r)x + W(\omega^r)y = h(\omega^r) \\ & y \geq 0. \end{array} \tag{$\mathcal{P}_r$}$$

To determine the infeasibility of this problem, all that matters to see is whether the constraints are satisfied, if not, one must be occur for $y \geq 0$:

$$M(\omega^r)x + W(\omega^r)y \neq h(\omega^r) \tag{3.4}$$

this fact can be corroborated by introducing an artificial variable $v \in \mathbb{R}$ and measure the amount by which the restrictions can be violated, for this reason, the following linear programming problem is formulated :

$$\begin{array}{rl} z_r = \min & e^\top v^+ + e^\top v^- \\ s.t. & M(\omega^r)x + W(\omega^r)y + Iv^+ - Iv^- = h(\omega^r) \\ & y \geq 0, v^+ \geq 0, v^- \geq 0. \end{array} \tag{3.5}$$

where $e^\top = (1, \ldots, 1)$ and I is the identity matrix with its appropriate dimension, and the artificial variable v was replaced by two nonnegative variables v^+, v^-. If $z_r = 0$, it indicates that there is no violation of the constraints and therefore the problem $(\mathcal{P}_r)$ is feasible. If $z_r > 0$ then it means that there is a violation of the constraints and therefore the problem is infeasible, in this case we can add a feasibility cut to a problem called the relaxed master problem. Remember, though, this process only reveals feasibility constraints that were violated with the current value of x, so when we re-solve the relaxed master problem with these new feasibility cuts we will get a new optimal value of x and we will have to repeat this process. To see the feasible cut to aggregate, considere the dual problem of (3.5):

$$\begin{aligned} a_r = \max \quad & (\sigma_r)^\nu (h(\omega^r) - M(\omega^r)x) \\ s.t. \quad & (\sigma_r)^\top W(\omega^r) \le 0 \\ & |\sigma_r| \le e \end{aligned} \tag{3.6}$$

By strong duality, if the problem $(\mathcal{P}_r)$ is feasible, the optimal value a_r has to be zero, else the value of a_r result positive. We want to avoid the case $a_r > 0$, so we must impose the condition $a_r \geq 0$, i.e., if x is the first-stage taken, and $(\sigma_r)^\nu$ is the optimal value of (3.6), the constraint we need to add into the relaxed master problem is therefore

$$(\sigma_r)^\top (h(\omega^r) - M(\omega^r)x) \le 0 \tag{3.7}$$

With

$$D_l = (\sigma_r)^\top M(\omega^r)x, \qquad d_l = (\sigma_r)^\top h(\omega^r) \tag{3.8}$$

Note that this restriction is added for the x taken, if a new x is taken again, the process must be repeated. The process described is when there is a scenario for which some sub problem is infeasible. However, it is possible that all subproblems are feasible, we will cover that below.

3.1.2 Optimality cuts

Suppose the first-stage decision is an arbitrary $x \in K_1 = \{Ax = b, x \geq 0\}$. If all feasible cuts have already been generated, it is possible that after this step, the optimal solution is still not found. In order to reach an optimal solution rather than a merely feasible one we also need to generate optimality cuts, which send information back to the first stage problem about how to make the first-stage decision in such a way as to be optimal. Looking at the problem

$$\begin{aligned} \min_x \quad & c^\top x + \theta \\ s.t \quad & V(x) \le \theta \\ & x \in K_1 \cap K_2 \end{aligned} \tag{3.9}$$

if we have the first-stage decision x^ν, for all $r = 1, \dots, R$, the problem

$$\begin{array}{rl} Q_r(x^\nu) = \min & q(\omega^r)^\top y \\ s.t. & M(\omega^r)x^\nu + W(\omega^r)y = h(\omega^r) \\ & y \geq 0. \end{array} \qquad (\mathcal{P}_r)$$

is feasible with optimal dual variable π_r^ν. Then

$$Q_r(x^\nu) = (\pi_r^\nu)^\top (h(\omega^r) - M(\omega^r)x^\nu) \qquad (3.10)$$

Moreover, by convexity of $Q_r(x)$, it follows from the subgradient inequality that

$$Q_r(x) \geq (\pi_r^\nu)^\top (h(\omega^r) - M(\omega^r)x) \qquad (3.11)$$

Hence, the expected second-stage value function $V(x)$ is given by

$$\mathcal{V}(x^\nu) = \mathbb{E}(Q_r(x^\nu)) = \sum_{r=1}^{R} p_r (\pi_r^\nu)^\top (h(\omega^r) - M(\omega^r)x^\nu) \qquad (3.12)$$

and follow from (3.11)

$$V(x) \geq \sum_{r=1}^{R} (\pi_r^\nu)^\top p_r (h(\omega^r) - M(\omega^r)x) \qquad (3.13)$$

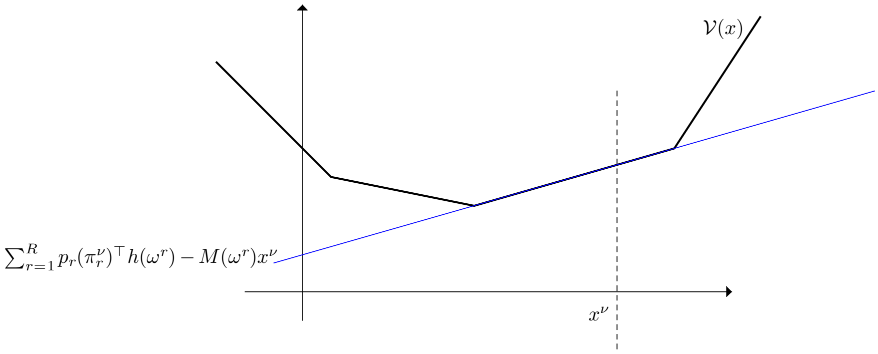
$\mathcal{V}(x)$
$\sum_{r=1}^{R} p_r(\pi_r^\nu)^\top h(\omega^r) - M(\omega^r)x^\nu$
x^ν

Figure 3.1: Approximation of the value function $\mathcal{V}(x)$ at iteration x^ν.

By $\theta \geq V(x)$, it follows that a pair (x, θ) is feasible only if

$$\theta \geq \sum_{r=1}^{R} (\pi_r^\nu)^\top p_r (h(\omega^r) - M(\omega^r)x)$$

On the other hand, if (x^ν, θ^ν) is the current optimal value for (3.2), it follows that $V(x^\nu) = \theta^\nu$. So, this is a terminal criterion. If this is not the case, then it should be added the **optimality cuts**

$$E_s = \sum_{r=1}^{R} p_r (\pi_r^\nu)^\top M(\omega^r) \tag{3.14}$$

$$e_s = \sum_{r=1}^{R} p_r (\pi_r^\nu)^\top h(\omega^r) \tag{3.15}$$

In summary:

L-Shaped Algorithm

0. Set $\nu = s = m = 0$ for all $r = 1, \ldots, R$.

1. Solve the linear program

$$\begin{array}{rll} \min & c^\top x + \theta & \\ s.t. & Ax = b & \\ & E_l x + \theta \geq e_l & l = 1, \ldots, s \\ & D_l x \geq d_l & l = 1, \ldots, m \\ & x \geq 0, \theta_r \in \mathbb{R} & r = 1, \ldots, R \end{array} \tag{3.16}$$

   Let (x^ν, θ^ν) be an optimal solution. If no inequality constraint is present for θ is set equal to $-\infty$. If not, go to step 2.

2. Check if $x^\nu \in K_2$: For $r = 1, \ldots, R$ solve the linear program

$$\begin{array}{rl} \min\limits_{y, v^+, v^-} & z_r = e^\top v^+ + e^\top v^- \\ s.t. & W(\omega^r) y + I v^+ - I v^- = h(\omega^r) - M(\omega^r) x^\nu \\ & y \geq 0, v^+ \geq 0, v^- \geq 0 \end{array} \tag{3.17}$$

   where $e^\top = (1, \ldots, 1)$ and I is the identity matrix. Let σ^ν the dual simplex variable associated for this problem. For r such that $z_r > 0$ define

$$D_{m+1} = (\sigma^\nu)^\top M(\omega^r) \tag{3.18}$$

$$d_{m+1} = (\sigma^\nu)^\top h(\omega^r) \tag{3.19}$$

   to generate the **feasibility cut** $D_{m+1} x \geq d_{m+1}$, add to the constraint set. Set $m = m + 1$, update $\nu = \nu + 1$ and return to Step 1. If $z_r = 0$ for all $r = 1, \ldots, R$, go to step 3.

3. For $r = 1, \dots, R$ solve the linear program

$$\begin{aligned} Q_r(x^\nu) = \min_y \quad & q(\omega^r)^\top y \\ s.t. \quad & W(\omega^r)y = h(\omega^r) - M(\omega^r)x^\nu \\ & y \geq 0 \end{aligned} \tag{$\mathcal{P}_r$}$$

Let π_r^ν the dual optimal solution of $(\mathcal{P}_r)$. Define

$$E_{s+1} = \sum_{r=1}^{R} p_r(\pi_r^\nu)^\top M(\omega^r) \tag{3.20}$$

$$e_{s+1} = \sum_{r=1}^{R} (\pi_r^\nu)^\top h(\omega^r) \tag{3.21}$$

Let $w^\nu = e_{s+1} - E_{s+1}x^\nu$. If $\theta^\nu \geq w^\nu$, stop x^ν is an optimal solution. Else, Set $s = s + 1$, update $\nu = \nu + 1$ and return to Step 1.

3.2 Multicut L-shaped Algorithm

It is a variation of the L-shaped method, in which the master problem is the following

$$\begin{array}{lll} \min & c^\top x + \sum_{r=1}^{R} \theta_r & \\ s.t. & Ax = b & \\ & E_{l(r)}x + \theta_r \geq e_{l(r)} & l(r) = 1, \dots, s_r \\ & D_l x \geq d_l & l = 1, \dots, m \\ & x \geq 0, \theta_r \in \mathbb{R} & r = 1, \dots, R \end{array} \tag{3.22}$$

The formulation differs in the construction of the optimal cuts are built separately for each possible scenario.

Multicut L-shaped

0. Set $\nu = s_r = N_r = 0$ for all $r = 1, \dots, R$.

1. Solve the linear program

$$\begin{array}{lll} \min & c^\top x + \theta & \\ s.t. & Ax = b & \\ & E_{l(r)}x + \theta \geq e_{l(r)} & l(r) = 1, \dots, s_r \\ & D_l x \geq d_{l)} & l = 1, \dots, m \\ & x \geq 0, \theta_r \in \mathbb{R} & r = 1, \dots, R \end{array} \tag{3.23}$$

Let $(x^\nu, \theta_1^\nu, \dots, \theta_R^\nu)$ be an optimal solution. If no inequality constraint is present for some r, then θ_r^ν is set equal to $-\infty$. If not, go to step 2.

2. Check if $x^\nu \in K_2$: For $r = 1, \dots, R$ solve the linear program

$$\begin{aligned} \min_{y,v^+,v^-} \quad & z_r = e^\top v^+ + e^\top v^- \\ s.t. \quad & W(\omega^r)y + Iv^+ - Iv^- = h(\omega^r) - M(\omega^r)x^\nu \\ & y \geq 0, v^+ \geq 0, v^- \geq 0 \end{aligned} \tag{3.24}$$

where $e^\top = (1, \dots, 1)$ and I is the identity matrix. Let σ^ν the dual simplex variable associated for this problem. For r such that $z_r > 0$ define

$$D_{m+1} = (\sigma^\nu)^\top M(\omega^r) \tag{3.25}$$

$$d_{m+1} = (\sigma^\nu)^\top h(\omega^r) \tag{3.26}$$

to generate the **feasibility cut** $D_{m+1}x \geq d_{m+1}$, add to the constraint set. Set $m = m + 1$, update $\nu = \nu + 1$ and return to Step 1. If $z_r = 0$ for all $r = 1, \dots, R$, go to step 3.

3. For $r = 1, \dots, R$ solve the linear program

$$\begin{aligned} Q_r(x^\nu) = \min_y \quad & q(\omega^r)^\top y \\ s.t. \quad & W(\omega^r)y = h(\omega^r) - M(\omega^r)x^\nu \\ & y \geq 0 \end{aligned} \tag{$\mathcal{P}_r$}$$

Let π_r^ν the dual optimal solution of $(\mathcal{P}_r)$. For all $r = 1, \dots, R$ such that

$$\theta_r^\nu < (\pi_r^\nu)^\top (h(\omega^r)) - (\pi_r^\nu) M(\omega^r) x^\nu \tag{3.27}$$

Define

$$E_{s_r+1} = p_r (\pi_r^\nu)^\top M(\omega^r) \tag{3.28}$$

$$e_{s_r+1} = p_k (\pi_r^\nu)^\top h(\omega^r) \tag{3.29}$$

and construct the **optimality cuts**

$$L_r(x) = e_{s_r+1} - E_{s_r+1} x \tag{3.30}$$

Set $s_r = s_r + 1$, update $\nu = \nu + 1$ and return to Step 1. If (3.27) does not hold for any $r = 1, \dots, R$, stop; x^ν is an optimal solution to the deterministic equivalent problem.

3.3 Nested L-shaped Algorithm

Nested L-Shaped algorithm, as the name hints, takes the idea of L-Shaped decomposition and applies it recursively over a tree structure, viewing the tree as set of nested two-stage problems

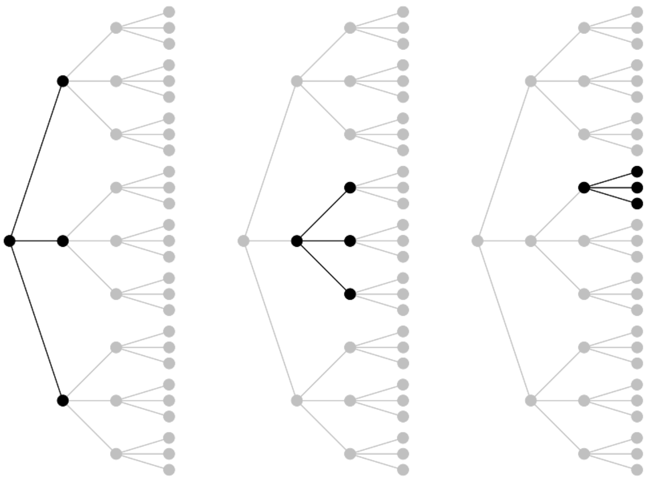

Figure 3.2: Multistage problem as a nested series of two-stage problems to which we can apply L-shaped decomposition.

Formulation

We can look at each node with its children as a two-stage problem. We should also note that each node is a child of some other node, except the first stage node. The difference with the L-shaped method is that now we must connect more information passing from node to node. This passing of information makes the iterations a bit more difficult.

In a two-stage problem, the number of scenarios is equal to the number of leaves. Then, for a T-stage stochastic program we can formulate the two-stage subproblems at each node as :

1. For the root node, the subproblem is:

$$\begin{array}{lll} \min & c^\top x_0^1 + \theta_0^1 & \\ & Ax_0^1 = b_0 & \\ & D^{1,j} x_0^1 \geq d_t^{1,j} & j = 1, \ldots, r_0^1 \\ & E_t^{1,j} x_0^1 + \theta_0^1 \geq e_t^{1,j} & j = 1, \ldots, m_0^1 \\ & x_0^1 \geq 0 & \end{array} \qquad (\mathcal{P}_{1,0})$$

2. for $t = 1, \ldots, T-2$ and $k = 1, \ldots, \mathcal{K}^{t-1}$ the scenarios at stage t :

$$\begin{array}{llll} \min & c^\top x_t^k + \theta_t^k & & \\ & W_t^k x_t^k = h_t^k - M_{t-1}^k x_{t-1}^{a(k)} & & (\pi_t^k) \\ & D^{k,j} x_t^k \geq d_t^{k,j} & j = 1, \ldots, r_t^k & (\rho_t^k) \\ & E_t^{k,j} x_t^k + \theta_t^k \geq e_t^{k,j} & j = 1, \ldots, m_t^k & (\sigma_t^k) \\ & x_t^k \geq 0 & & \end{array} \qquad (\mathcal{P}_{k,t})$$

where $a(k)$ is the ancestor scenario of k at stage $t-1$, $x_{t-1}^{a(k)}$ is the current solution from that scenario.

3. For the last stage $T-1$ there is no cost-to-go function, so there are no cut restrictions, that is :

$$\begin{array}{ll} \min & c^\top x_{T-1}^k \\ & W_{T-1}^k x_{T-1}^k = h_T^k - M_{T-2}^k x_{T-2}^{a(k)} \\ & x_{T-1}^k \geq 0 \end{array} \qquad (\mathcal{P}_{k,T})$$

Basic Idea of Implementation

We proceed as follows. First we solve the root node problem (with no constraints from subsequent stage sub-problems). We then take this optimal root decision and solve the sub-problems at the second stage (without any constraints coming from their subsequent stage sub-problems). These will generate feasibility or optimality cuts. So far, so like L-shaped Algorithm. After solving the second stage problems we now have a choice. We can go back and solve the root node problem again using the cut or cuts that have been generated from the second stage problems or we can take the second stage optimal solutions that we have found (for the second stage problems that proved to be feasible) and move to the third stage, solving the sub-problems there. If we go to the third stage we will solve the sub-problems there (again without any constraints coming from their sub-problems) and generate some cuts for the second stage problems. We could then, if we wanted, move back up to the second stage and re-solve there or move on to the fourth stage. This process can (if desired) continue until we get to the leaves of the tree when it is no longer possible to move any further down and we have to start moving back up the tree. A nice way to think of the algorithm is that solutions are passed down the tree from parent nodes to children whilst cuts are passed back up the tree from children to parents.

But, throughout this procedure a question arises after the resolution of each subproblem: What should we do with the information obtained (dual and primal solutions)? The answer to this question is very important, because if we choose to use the dual information of the subproblem at the current node (to optimality cuts), then the next subproblem to be solved is from the parent node in the previous stage. But, if we choose to use primal information (when the subproblem is feasible), the next subproblem to be solved is all of the children nodes.

Sequencing Heuristics

Many alternative strategies are possible in this algorithm in terms of determining the next subproblem to solve. There are three main strategies (heuristics) that are used in deciding which way to move and when in the Nested L-shaped algorithm. Such heuristics are known as sequencing heuristics or sequencing protocols

1. **Fast-forward (FF)**: Aims to descend the tree whenever possible. So, at any point where a subproblem is feasible and we get a solution we will descend to the children nodes taking the solution with us until we either reach the leaves of the tree or infeasibility on all sub-problems under consideration. We say that the flow of information moves in a forward direction (*DIR=FORE*)
2. **Fast-back (FB)**: Aims to go back up the tree as far and as fast as it can wherever possible. So, whenever we add a cut from a child problem we will re-solve the parent and, if this generates a new cut, go to its parent and re-solve that, until we reach the root. We say that the flow of information moves in a backward direction (*DIR=BACK*)
3. **Fast-forward-fast-back (FFFB)**: is a combination of the two methods above. In this case we have a switch that puts the algorithm in either descending or ascending mode. When in descending mode we follow the fast-forward scheme, going down the tree until we can go no further. When we can go no further we switch to ascending mode and then follow the fast-back scheme, ascending the tree until again we can go no further. We then switch back to descending mode.

The FFFB procedure was proposed by Eittrock with successs impmentations by Gassmann which also showed that FFFB procedure seems generally more efficient thant the alternatives.

Nested L-shaped with Fast-forward- fast-back protocole

Step 0. Solve the root node without cuts, i.e., $t = 0, k = 1, r_t^k = m_t^k = 0$ and add the constraint $\theta_t^k = 0$ for all t and k to $(\mathcal{P}_{k,t})$. Let *DIR=FORE*. Go to Step 1.

Step 1. Solve the current problem $(\mathcal{P}_{k,t})$.

a) If is infeasible at $t = 0$, then **STOP**; all the problem is infeasible. If is infeasible at $t > 1$, then let

$$r_{t-1}^{a(k)} = r_{t-1}^{a(k)} + 1$$

and let *DIR=BACK*. Let the infeasibility condition be obtained by a dual basic solution $\pi_t^k, \rho_t^k \geq 0$ such that

$$(\pi_t^k)^\top W_t^k + (\rho_t^k)^\top D_t^k \geq 0 \tag{3.31}$$

but

$$(\pi_t^k)^\top (h_t^k - T_{t-1}^k x_{t-1}^{a(k)}) + (\rho_t^k)^\top d_t^k > 0 \tag{3.32}$$

Then make

$$D_{t-1}^{a(k),r_{t-1}^{a(k)}} = (\pi_t^k) M_{t-1}^k, d_{t-1}^{a(k),r_{t-1}^{a(k)}} = (\pi_t^k)^\top h_t^k + (\rho_t^k)^\top d_t^k. \tag{3.33}$$

Let $t = t - 1$, $k = a(k)$ and return to Step 1.

b) If the problem $(\mathcal{P}_{k,t})$ is feasible, update the values of x_t^k, θ_t^k, and store the value of the complementary basic dual multipliers of the restriction in that order $(\pi_t^k, \rho_t^k, \sigma_t^k)$. If $k < \mathcal{K}^t$, let $k = k + 1$, and return to Step 1. Otherwise, $(k = \mathcal{K}^t)$, if $t = 0$, set *DIR=FORE*; if *DIR=FORE* and $t < T - 1$ (not yet in final nodes), let $t = t + 1$ and return. If $t = T - 1$ (final nodes), let *DIR=BACK*. Go to Step 2.

Step 2. If $t = 0$, let $t = t + 1$, $k = 1$ and go to Step 1. Otherwise, for all scenarios $j = 1, \ldots, \mathcal{K}^{t-1}$ at $t - 1$ compute

$$E_{t-1}^j = \sum_{k \in \mathcal{D}^t(j)} \frac{p_t^k}{p_{t-1}^j} (\pi_t^k)^\top M_{t-1}^k \tag{3.34}$$

and

$$e_{t-1}^j = \sum_{k \in \mathcal{D}^t(j)} \frac{p_t^k}{p_{t-1}^j} (\pi_t^k) \left[(\pi_t^k)^\top h_t^k + \sum_{i=1}^{r_t^k} (\rho_t^{ki})^\top d_t^{ki} + \sum_{i=1}^{m_t^k} (\sigma_t^{ki} e_t^{ki}) \right] \tag{3.35}$$

The current conditional expected value of all scenario problems in $\mathcal{D}^t(j)$ is then

$$\overline{\theta}_{t-1}^{j} = e_{t-1}^{j} - E_{t-1}^{j} x_{t-1}^{j}. \tag{3.36}$$

If the constraint $\theta_{t-1}^{j} = 0$ appears in $(\mathcal{P}_{j,t-1})$, then remove it and : let $m_{t-1}^{j} = 1$, and add the linear constraint with coefficients E_{t-1}^{j} and e_{t-1}^{j} to $(\mathcal{P}_{j,t-1})$. If $\overline{\theta}_{t-1}^{j} > \theta_{t-1}^{j}$, then let $m_{t-1}^{j} = m_{t-1}^{j} + 1$ and add the linear constraint with coefficients E_{t-1}^{j} and e_{t-1}^{j} to $(\mathcal{P}_{j,t-1})$. If $t = 1$ and no constraints are added to $(\mathcal{P}_{1,0})$, then stop with x_0^1 optimal. Otherwise, let $t = t - 1$, $k = 1$. If $t = 0$, let *DIR=FORE*. Go to Step 1.

3.3.1 Convergence Analysis

For the convergence of the Nested L-Shaped with the FFFB protocole, we will first demonstrate that all cuts generated are valid aproximation (specifically outer linearizations) of the feasible regions and for each value function $\mathcal{V}_t(x_{t-1})$.

By induction on t, suppose that all feasible cuts on $(\mathcal{P}_{k,t})$ generated by the algorithm for periods t or greater are valid. For $t = T$, no cuts are present so this is true for the last period. In this case, for any $\pi_t^k, \rho_t^k \geq 0$ such that

$$\left(\pi_t^k\right)^\top W_t + \left(\rho_t^k\right)^\top D_t^k \leq 0,$$

we must have

$$\left(\pi_t^k\right)^\top \left(h_t^k - M_{t-1}^k x_{t-1}^{a(k)}\right) + \left(\rho_t^k\right)^\top d_t^k \leq 0$$

to maintain feasibility. Because this is the cut added, these cuts are valid for $t-1$. Thus, the induction is proved.

Now, suppose the cuts in $(\mathcal{P}_{k,t})$ are an outer linearization of $\mathcal{V}_{t+1}^k\left(x_t^k\right)$ for t or greater and all k. In this case, for any $(\pi_k^t, \rho_k^t, \sigma_k^t)$ feasible in $(\mathcal{P}_{k,t})$ for t and k,

$$\left(\pi_k^t\right)^T \left(h_t^k - M_t^k x_{t-1}^{a(k)}\right) + \sum_{i=1}^{r_k} \left(\rho_t^{ki}\right)^\top d_t^{ki} + \sum_{i=1}^{m_k} \left(\sigma_t^{ki}\right)^\top e_t^{ki}$$

is a lower bound on $Q_t^{a(k)}\left(x_{t-1}^{a(k)}, k\right)$ for any $x_{t-1}^{a(k)}$, each k, and $a(k)$. Thus, we must have

$$\mathcal{V}_t^{a(k)}\left(x_{t-1}^{a(k)}\right) \geq \sum_{k \in \mathcal{D}^t(a(k))} \left(\frac{p_t^k}{p_{t-1}^{a(k)}}\right) \left(\left(\pi_t^k\right)^\top \left(h_t^k - M_t^k x_{t-1}^{a(k)}\right)\right.$$
$$\left. + \sum_{i=1}^{r_k} \left(\rho_t^{ki}\right)^\top d_t^{ki} + \sum_{i=1}^{s_k} \left(\sigma_t^{ki}\right)^\top e_t^{ki}\right)$$

which says that $\theta_{t-1}^{k} \geq -E_{t-1}^{a(k)} x_{t-1}^{a(k)} + e_{t-1}^{a(k)}$, as found in the algorithm. Thus, again, we achieve a valid cut on $\mathcal{V}_{t-1}^{a(k)}$ for any $a(k)$, completing the induction.

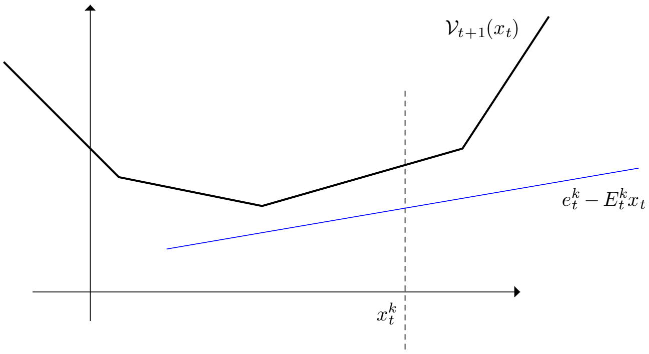

Figure 3.3: Unlike the L-shaped method, the Nested L-shaped method applied to a multi-stage problem can produce optimal cuts that are not support hyperplanes.

Now, suppose that the algorithm terminates. This can only happen if $(\mathcal{P}_{k,t})$ is infeasible for $t = 0$ or if each subproblem for $t = 1$ has been solved and no cuts are generated. In the former case, the problem is infeasible, because the cuts feasibility cuts are all outer linearizations of the feasible region. In the latter case, we must have $\theta_0 = \mathcal{V}_1(x_0)$, the condition for optimality.

For finiteness, proceed by induction. Suppose that at stage t, at most a finite number of cuts from stage $t+1$ to $T-1$ can be generated for each k at t. For $T-1$, this is again trivially true. Because at most a finite number of cuts are possible at each k, at most a finite number of basic solutions, $\left(\pi_t^k, \rho_t^k, \sigma_t^k\right)$, can be generated to form cuts for $a(k)$. Thus, at most a finite number of cuts can be generated for all $a(k)$ at $t-1$, again completing the induction.

Finally by noting that every iteration of Step 1 or 2 produces a new cut, we obtain that the nested decomposition algorithm converges to the optimal solution. Because there is only a finite number of possible cuts, the procedure stops finitely.

Chapter 4

Progressive-Hedging Algorithm

In this chapter we explain the progressive-hedging algorithm which was first introduced in [11] by Rockafellar and Wets. In contrast to the nested decomposition algorithm, progressive-hedging give us a complete decomposition of the problem (2.6) over the total number of scenarios. It will also be shown how we can obtain the progressive-hedging algorithm as a particular application of the proximal point algorithm.

4.1 Scenario Aggregation Principle

In subsection 2.2.2 the multistage stochastic linear programming with explicit nonanticipativity constraints was formulated as

$$\begin{aligned} \min_{X \in \mathcal{E}} \quad & \sum_{s \in S} p(s) f_s(X(s)) \\ s.t \quad & X \in \mathcal{C} \cap \mathcal{N} \end{aligned} \tag{4.1}$$

If we write

$$F(X) = \sum_{s \in S} p(s) f_s(X(s))$$

then the problem (4.1) can be reformulated to

$$\min F(X) \text{ subject to } X \in \mathcal{C} \cap \mathcal{N} \tag{$\mathcal{P}$}$$

A policy $X \in \mathcal{C} \cap \mathcal{N}$ will be called **feasible** (implementable and admissible) and the optimal policy is one that responds best among all feasible policies to the relative importance of the scenarios (weights $p(s)$).

The central question in this chapter is how to find an optimal feasible policy. We will show the scenario aggregation principle, which uses the solutions of some modified scenario subproblems to get closer and closer to the optimal solution. First, we must know how to convert a general policy X into an implementable policy. The idea is "homogenize" the different values for the scenarios in a scenario bundle at each stage using a wighted averaged.

Specifically from any policy X we obtain a implementable policy $\hat{X}$ doing for every stage t and for every $A \in \mathcal{A}_t$:

$$\hat{X}_t(s) = \frac{\sum_{\omega \in A} p(\omega) X_t(\omega)}{\sum_{\omega \in A} p(\omega)}, \forall s \in A \tag{4.2}$$

Clearly $\hat{X}$ is implementable : $\hat{X} \in \mathcal{N}$. The operator

$$\begin{array}{cccc} \psi & \mathcal{E} & \to & \mathcal{E} \\ & X & \mapsto & \psi(X) = \hat{X} \end{array} \tag{4.3}$$

defined by the expression (4.2) is called **aggregation operator** relative to the given information structure and weigths. With a little calculation it can be shown that ψ is a linear transformation such that $\psi^2 = \psi$.

Now, is introduced an inner product in the policy space $\mathcal{E}$ defined by

$$\langle X, Y \rangle_{\mathcal{E}} = \sum_{s \in S} p_s \langle X(s), Y(s) \rangle$$

where $\langle \cdot, \cdot \rangle$ is the canonical inner product in $\mathbb{R}^n$. This inner product over $\mathcal{E}$ gives us the following result which is fundamental

Proposition 4.1 *:*

1. *The agreggation operator $\psi : X \mapsto \hat{X}$ is the orthogonal projection on the subespace $\mathcal{N}$.*
2. *The operator $K = \mathrm{Id} - \psi$ is the orthogonal projection on the orthogonal complement of $\mathcal{N}$, wich we denote $\mathcal{M}$.*

Proof :

1. It will be shown that $\operatorname{Ker} \psi = \mathcal{N}^{\perp}$. Let $X \in \operatorname{Ker} \psi$ then $\psi X = \hat{X} = 0$.

Let $Y \in \mathcal{M}$:

$$\begin{aligned}\langle X, Y\rangle_{\mathcal{E}} &= \sum_{s\in S} p_s \langle X(s), Y(s)\rangle \\ &= \sum_{t=1}^{T}\sum_{s\in S} p(s) X_t(s)\cdot Y_t(s) \\ &= \sum_{t=1}^{T}\sum_{A\in\mathcal{A}_t} p(A) Y_t(A)\cdot X_t(A) \\ &= 0\end{aligned}$$

where $p(A) = \sum_{\omega\in A} p(\omega)$, $Y_t(A) = \sum_{\omega\in A} Y_t(\omega)/p(A)$ and $X_t(A) = \sum_{\omega\in A} X_t(\omega)/p(A)$

2. If $X \in \mathcal{M}$ then $\langle X, \hat{X}\rangle_{\mathcal{E}} = 0$ wich means that

$$\sum_{t=1}^{T}\sum_{A\in A} p(A) X_t(A)\cdot X_t(A) = 0$$

then $X_t(A) = 0$ for all $A \in \mathcal{A}_t$ at each stage t. Hence $X \in \operatorname{Ker}\psi$. Thus, Y is the orthogonal projection of the policy space $\mathcal{E}$ over $\mathcal{N}$. Immediately from this it follows that $K = \operatorname{Id} - \psi$ is is the orthogonal projection on the subspace $\mathcal{M}$. □

Finally we have the following equivalences

$$\mathcal{M} = \mathcal{N}^{\perp} = \{X \in \mathcal{E} : \psi X = 0\}$$
$$\mathcal{N} = \{X \in \mathcal{E} : KX = 0\}$$

Thus a policy X is implementable if and only if it satisfies the linear constraint equation $KX = 0$. Therefore problem $(\mathcal{P})$ is reformulated to

$$\min F(X) \text{ s.t } X \in \mathcal{C}, KX = 0 \qquad (\mathcal{P})$$

The challenge then is to find X^* without going beyond the tools that are available

1. We can quickly compute for each X, the projection onto $\mathcal{N}$: $\hat{X} = \psi X$ and the projection onto $\mathcal{M}$: $KX = X - \hat{X}$.

2. We can solve a certain modified version of the scenario subproblems $(\mathcal{P}_s)$, specifcally

$$\min f_s(x) + \langle x, w\rangle + \frac{1}{2} r|x - \hat{x}|^2 \text{ over all } x \in C(s) \qquad (\hat{\mathcal{P}}_s(\hat{x}, w, r))$$

$w \in \mathbb{R}^n$ will be the price vector and $r > 0$ a penalty paramenter.

The motivation to form these modified subproblems comes in part from a Lagrangian function associated with problem $(\mathcal{P})$. The ordinary Lagrangian $L(X,Y)$ for the problem $(\mathcal{P})$ is

$$L(X,Y) = F(X) + \langle KX, Y\rangle \text{ for } X \in \mathcal{C}, Y \in \mathcal{E}$$

Since K is an orthogonal projection, we have $\langle KX, Y\rangle = \langle X, KY\rangle$. Writing $KY = W \in \mathcal{M}$, is conveniently to define

$$L(X,W) = F(X) + \langle X, W\rangle \text{ for } X \in \mathcal{C}, W \in \mathcal{M} \tag{4.4}$$

as the actual Lagrangian for $(\mathcal{P})$. But more powerfull in many ways (existence of solutions and numerical results) is the *augmented Lagrangian*

$$L_r(X,W) = F(X) + \langle X, W\rangle + \frac{1}{2}r\|KX\|^2, \text{ for } X \in \mathcal{C}, W \in \mathcal{M}, r > 0 \tag{4.5}$$

where W is called **information price system**, because it will help us determine the changes in prices for an optimal X policy found. The augmented Lagrangian provide us the basis of the solution procedure, for example the method of multipliers in which the iterates are generated by

$$X^\nu \in \operatorname{argmin} L_r(X, W^{\nu-1}) \tag{4.6}$$

$$W^\nu(s) = W^{\nu-1}(s) + r\left[X^\nu(s) - \hat{X}^\nu(s)\right], s \in S \tag{4.7}$$

The problem now is how to generate the X^ν iterates, the difficulty lies in the fact that the term $\|KX\|^2$ is not decomposable into separate terms for each scenario. But the implementation of the aggregation operator makes it possible to decompose (4.6) into modified scenario subproblems, since

$$\begin{aligned}\|KX\|^2 &= \langle X - \hat{X}, X - \hat{X}\rangle_{\mathcal{E}} \\ &= \sum_{s\in S} \langle X(s) - \hat{X}(s), \langle X(s) - \hat{X}(s)\rangle \\ &= \sum_{s\in S} \|X(s) - \hat{X}(s)\|^2\end{aligned}$$

Replacing this result on (4.5) we can write:

$$L_r(X, W^\nu) = \sum_{s\in S} p(s)\left(f_s(X(s)) + \langle X(s), W^\nu(s)\rangle + \frac{1}{2}r\|X(s) - \hat{X}^\nu(s)\|\right)$$

Thus, we obtain the following algorithm called **Progressive Hedging**

Progressive Hedging Algorithm

Step 0. Initializate : $r > 0$, $W^0 = 0 \in \mathcal{M}$, $\hat{X}^0 = J(X^0) \in \mathcal{N}$ (where X^0 is a policy such that $X^0(s)$ is for each s an optimal solution to the unmodified scenario subproblem $(\mathcal{P}_s)$), and set $\nu = 0$.

Step 1. Calculate the optimal solution $X^{\nu+1}$ of the subproblem

$$\min_{X \in \mathcal{C}} \left[F(X) + \langle X, W^\nu \rangle_{\mathcal{E}} + \frac{1}{2} r \| X - \hat{X}^\nu \|^2_{\mathcal{E}} \right] \qquad (\mathcal{P}^\nu)$$

This descomposing into solving (aproximately) for each $s \in S$ the subproblem

$$\min_{X(s) \in \mathcal{C}(s)} \left[f_s(X(s)) + \langle X(s), W^\nu(s) \rangle + \frac{1}{2} r \| X(s) - \hat{X}^\nu(s) \|^2 \right] \quad (\mathcal{P}^\nu_s)$$

in order to get $X^{\nu+1}(s)$. The policy $X^{\nu+1}$ will again be admissible but not necessarily implementable.

Step 2. Calculate the policy $\hat{X}^{\nu+1} = J(X^{\nu+1})$, wich is implementable but not necessarily admissible. (If ever one wishes to stop, this policy $\hat{X}^{\nu+1}$ is to be offfered as the best substitute yet available for a solution to $(\mathcal{P})$).

Step 3. Update the price system

$$W^{\nu+1} = W^\nu + r[X^{\nu+1} - \hat{X}^{\nu+1}] \qquad (4.8)$$

Return to step 1 with $\nu = \nu + 1$.

4.1.1 Projected Problem

If we were to start from a X^0 in which $X^0(s)$ is for each s an optimal solution to the unmodified scenario subproblem $(\mathcal{P}_s)$, which is the typical beginning for all scenario analysis, the corresponding implementable policy $\hat{X}^0 = \psi X^0$ might be contemplated as a kind of solution to the underlying problem. There is no guarantee, however, that $\hat{X}^0$ will inherit from X^0 the property of admissibility. Even if $\hat{X}^0$ is admissible as well as implementable, therefore feasible, the sense in which it might be regarded as "optimal" needs to be clarified. In fact $\hat{X}^0$ is an optimal solution to the following problem

$$\begin{array}{rl} \min & \hat{F}(Y) \\ s.t & Y \in \hat{C} \end{array} \qquad (\hat{\mathcal{P}})$$

where $\hat{C} = \{Y \in \mathcal{N} : \exists X \in \mathcal{C} \text{ with } \psi X = Y\}$ and $\hat{F}(Y) = \min\{F(X) : X \in \mathcal{C}, \psi X = Y\}$. This problem is called "projected problem" because the set

$\hat{C}$ is precisely the projection of $\mathcal{C}$ under the aggregation operator ψ.

Indeed, $\hat{X}^0 \in \hat{C}$ because $\psi X^0 = \hat{X}^0$ where $X^0(s) \in \mathcal{C}(s)$ is the optimal solution to $(\mathcal{P}_s)$. By definition $\hat{F}(\hat{X}^0) = F(X^0)$. Suppose now that $Y \neq \hat{X}^0 \in \hat{C}$ exists such that $F(\hat{Y}) < \hat{F}(\hat{X}^0)$, then there would be $\overline{X} \in \mathcal{C}$ such that $\psi\overline{X} = Y$ and $F(\hat{Y}) = F(\overline{X})$, which is a contradiction because it would have to

$$F(X^0) \leq F(\overline{X}) < F(X^0)$$

So, what the progressive-hedging algorithm does is trace a path from $\hat{X}^0$ to X^* solving the modified subproblems $(\mathcal{P}_s^\nu)$. An advantage of this approach is that even if we do not pursue the search until $\hat{X}^\nu$ converges to X^* ,we always have at hand a solution estimate that is better that just $\hat{X}^0$ or any other policy that could reliably be gleaned from scenario analysis as practiced until now this will be demonstrated in the convergence section.

4.2 A view of PH from Monotone Operators

We now proceed to describe the progressive-hedging algorithm in the convex case (and therefore includes the linear case) as an application of the Douglas-Rachford splitting algorithm and the Spingarn's partial inverse method.

Recall that the problem $(\mathcal{P})$ to be solved is the following

$$\min F(X) \text{ over } X \in \mathcal{C} \cap \mathcal{N} \qquad (\mathcal{P})$$

Suppose that for each $s \in S$ the function f_s is a proper convex l.s.c. function and $C(s)$ is a convex set. Furthermore suppose that $\mathrm{ri}(\mathcal{C} \cap \mathcal{N}) \neq \emptyset$.

4.3 Douglas-Rachford splitting

By property of subdifferential (first order optimality condition), $(\mathcal{P})$ reduces to the following problem

$$\text{find } X^* \text{ such that } 0 \in \partial F(X^*) + N_{\mathcal{C}\cap\mathcal{N}}(X^*) \qquad (4.9)$$

Since $\mathrm{ri}(\mathcal{C} \cap \mathcal{N}) \neq \emptyset$ then we can write

$$\begin{aligned} \partial F(X) + N_{\mathcal{C}\cap\mathcal{N}}(X) &= \partial F(X) + N_{\mathcal{C}}(X) + N_{\mathcal{N}}(X) \\ &= \partial F(X) + \partial I_{\mathcal{C}}(X) + N_{\mathcal{N}}(X) \\ &= \partial(F(X) + I_{\mathcal{C}}(X)) + N_{\mathcal{N}}(X) \end{aligned}$$

The last equality is due to the fact that $F(X)$ is convex since it is a convex combination of convex functions, and $I_{\mathcal{C}}(X)$ is convex due to $C(s)$ is convex for all $s \in S$. Hence (4.9) is reformulated to

$$\text{find } X^* \text{ such that } 0 \in \partial \tilde{F}(X^*) + N_{\mathcal{N}}(X^*) \tag{4.10}$$

where $\tilde{F} = F + I_C$. Using the following notation

$$A(X) = \partial \tilde{F}(X) \quad \text{and} \quad B(X) = N_{\mathcal{N}}(X) \tag{4.11}$$

Under the assumption that all f_s are proper convex l.s.c. functions, the operators A and B are maximally monotonous. Finally, solving $(\mathcal{P})$ is equivalent to

$$\text{find } X^* \text{ such that } 0 \in A(X^*) + B(X^*) \tag{4.12}$$

According to Douglas Rachford splitting algorithm, we will have

$$X^* \text{ solve } (\mathcal{P}) \Leftrightarrow X^* = J_{\gamma B}(Z^*), \text{ with } Z^* \in \operatorname{Fix} R_{\gamma A} R_{\gamma B}$$

Since $R_{\gamma A} R_{\gamma B}$ is nonexpansive (as a composition of nonexpansive operators) but not necessarily firmly nonexpansive, the Banach-Picard iteration may fail to produce a point in $\operatorname{Fix} R_{\gamma A} R_{\gamma B}$. Therefore, we use the Krasnosel'skii-Mann iterations to the firmly nonexpansive operator $\frac{1}{2} R_{\gamma A} R_{\gamma B} + \frac{1}{2} \operatorname{Id}$. This give the following sequence

$$Z^{\nu+1} = \frac{1}{2} R_{\gamma A} R_{\gamma B}(Z^\nu) + \frac{1}{2} Z^\nu \tag{4.13}$$

which converges to $Z^* \in \operatorname{Fix} R_{\gamma A} R_{\gamma B}$ to then have $X^* = J_{\gamma B}(Z^*)$. The following proposition tells us what is the correspondence rule of the operators $R_{\gamma A}$ and $R_{\gamma B}$ to later replace it in (4.13) and get the Progressive Hedging Algorithm.

Proposition 4.2 *The reflected operators of A and B have the following expressions: Let $Z \in \mathcal{E}$*

(i) $R_{\gamma A}(Z) = 2U - Z$, *with $U \in \mathcal{E}$ a policy given by*

$$U(s) = \arg \min_{y \in C(s)} \left\{ f_s(y) + \frac{1}{2\gamma} \|y - Z(s)\|^2 \right\}, \text{ for all } s \in S$$

(ii) $R_{\gamma B}(Z) = 2V - Z$, *with $V \in \mathcal{E}$ a policy given by*

$$V_t(s) = \sum_{\omega \in A} p_\omega Z_t(\omega) / \sum_{\omega \in A} p_\omega, \forall s \in A \in \mathcal{A}_t, \forall t$$

(According to the definition of the aggregation operator, V is the orthogonal projection of Z on the space $\mathcal{N}$).

Proof (i) Let $U = J_{\gamma\partial\tilde{F}}(Z)$, since f_s and $I_{C(s)}$ are convex l.s.c. then $\tilde{F}$ is convex l.s.c and the resolvent is the proximal operator, hence

$$\begin{aligned}U &= \operatorname{prox}_{\gamma\tilde{F}}(Z)\\ &= \arg\min_{Y\in\mathcal{E}}\left\{\tilde{F}(Y) + \frac{1}{2\gamma}\|Y-Z\|^2_{\mathcal{E}}\right\}\\ &= \arg\min_{Y\in\mathcal{E}}\left\{F(Y) + I_C(Y) + \frac{1}{2\gamma}\|Y-Z\|^2_{\mathcal{E}}\right\}\\ &= \arg\min_{Y\in C}\left\{F(Y) + \frac{1}{2\gamma}\|Y-Z\|^2_{\mathcal{E}}\right\}\end{aligned}$$

by the inner product defined in $\mathcal{E}$, we obtain that U is a policy such that

$$U(s) = \arg\min_{y\in C(s)}\left\{f(y) + \frac{1}{2\gamma}\|y - Z(s)\|^2\right\},\ \text{for all } s\in S$$

(ii) Since $\mathcal{N}$ is a vector subspace in a finite dimensional space, then it is closed and convex. According to example 23.4, if $V = J_{\gamma N_{\mathcal{N}}}(Z)$ we will have that $J_{\gamma N_{\mathcal{N}}}(Z) = \psi(Z)$, i.e, the ortogonal projection Z on the space $\mathcal{N}$

$$V_t(s) = \sum_{\omega\in A} p_\omega Z_t(\omega) / \sum_{\omega\in A} p_\omega, \forall s\in A\in\mathcal{A}_t, \forall t$$

□

Now, we apply these expressions, in the iteration (4.13) we obtain for all $\nu = 1, 2, \ldots$

$$\begin{aligned}V_t^\nu(s) &= \sum_{\omega\in A} p_\omega Z_t^\nu(\omega) / \sum_{\omega\in A} p_\omega, \forall s\in A\in\mathcal{A}_t, \forall t\\ \mathcal{T}^\nu &= R_{\gamma B}(Z^\nu) = 2V^\nu - Z^\nu = V^\nu - \gamma W^\nu,\ \text{with } W^\nu = (Z^\nu - V^\nu)/\gamma\\ U^{\nu+1}(s) &= \arg\min_{y\in C(s)}\left\{f(y) + \frac{1}{2\gamma}\|y - \mathcal{T}^\nu(s)\|^2\right\}, \forall s\in S\\ Z^{\nu+1} &= \frac{1}{2}(2U^{\nu+1} - \mathcal{T}^\nu) + \frac{1}{2}Z^\nu\end{aligned}$$

Let us reorganize the equations and eliminate intermediate variables. In particular we use the fact that, provided that the algorithm is intialized with $V^0 \in \mathcal{N}$

and $W^0 \in \mathcal{M}$ we have

$$\begin{aligned} U^{\nu+1}(s) &= \arg\min_{y\in C(s)} \left\{ f(y) + \frac{1}{2\gamma}\|y - V^\nu(s) + \gamma W^\nu(s)\|^2 \right\}, \forall s \in S \\ V^{\nu+1}t(s) &= \sum_{\omega\in A} p_\omega U_t^\nu(\omega) / \sum_{\omega\in A} p_\omega, \forall s \in A \in \mathcal{A}_t, \forall t \\ W^{\nu+1} &= W^\nu + \frac{1}{\gamma}(U^{\nu+1} - V^{\nu+1}) \end{aligned}$$

Using the following equivalence

$$\begin{aligned} \|y - V^\nu(s) + \gamma W^\nu(s)\|^2 &= \langle y - V^\nu(s) + \gamma W^\nu(s), y - V^\nu(s) + \gamma W^\nu(s)\rangle \\ &= \frac{1}{2\gamma}\|y - V^\nu(s)\|^2 + \frac{2}{2\gamma}\langle \gamma W^\nu(s), y - V^\nu(s)\rangle + \frac{1}{2\gamma}\|W^\nu\|^2 \end{aligned}$$

and since the last term of this sum is a constant in each iteration, then we obtain

$$\begin{aligned} U^{\nu+1}(s) &= \arg\min_{y\in C(s)} \left\{ f(y) + \langle W^\nu(s), y - V^\nu(s)\rangle + \frac{1}{2\gamma}\|y - V^\nu(s)\|^2 \right\}, \forall s \in S \\ V^{\nu+1}t(s) &= \sum_{\omega\in A} p_\omega U_t^\nu(\omega) / \sum_{\omega\in A} p_\omega, \forall s \in A \in \mathcal{A}_t, \forall t \\ W^{\nu+1} &= W^\nu + \frac{1}{\gamma}(U^{\nu+1} - V^{\nu+1}) \end{aligned}$$

Which is the progressive hedging algorithm stated in chapter 2, with the notations $y = X(s)$, $U = X$, $V = \hat{X}$ and the value of $\gamma = 1/r > 0$. That is:

$$\begin{aligned} X^{\nu+1}(s) &= \arg\min_{y\in C(s)} \left\{ f(y) + \langle W^\nu(s), y - V^\nu(s)\rangle + \frac{1}{2\gamma}\|y - V^\nu(s)\|^2 \right\}, \forall s \in S \\ \hat{X}_t^{\nu+1}(s) &= \sum_{\omega\in A} p_\omega U_t^\nu(\omega) / \sum_{\omega\in A} p_\omega, \forall s \in A \in \mathcal{A}_t, \forall t \\ W^{\nu+1} &= W^\nu + \frac{1}{\gamma}(X^{\nu+1} - \hat{X}^{\nu+1}) \end{aligned}$$

4.4 Spingarn partial inverse

Consider the problem $(\mathcal{P})$ reformulated as follows

$$\min \tilde{F}(X) \text{ s.t } X \in \mathcal{N} \tag{4.14}$$

Let X an optimal solution of (4.14) such that satisfies the constraint qualification of Theorem 4.1, then exists $-W^* \in \partial\tilde{F}$ with $W^* \in \mathcal{M} = \mathcal{N}^\perp$. According to the method of partial inverses of Spingarn, we can formulate the following problem

$$\text{find } X^* \in \mathcal{N} \text{ and } W^* \in \mathcal{M} \text{ such that } -W^* \in \partial[F(X^*) + I_C(X^*)] \quad (4.15)$$

starting from $V^0 \in \mathcal{N}$ and $U^0 \in \mathcal{M}$ we get the following iterations for $\nu = 1, 2, 3, \ldots$

$$\begin{aligned} Y^{\nu+1} &= J_{\gamma\partial\tilde{F}}(V^\nu - \gamma U^\nu) \\ V^{\nu+1} &= P_{\mathcal{N}}(Y^{\nu+1}) \\ \gamma Z^{\nu+1} &= \gamma U^\nu + X^\nu + Y^{\nu+1} \\ U^{\nu+1} &= P_{\mathcal{M}}(Z^{\nu+1}) \end{aligned}$$

We know that the resolvent is the proximal operator, $P_{\mathcal{N}} = \psi$ and $P_{\mathcal{M}} = K$. Then for $\nu = 1, 2, \ldots$ we have

$$\begin{aligned} Y^{\nu+1} &= \arg\min_{Y \in \mathcal{C}} \left\{ F(Y) + \langle Y, U^\nu \rangle_{\mathcal{E}} + \frac{1}{2\gamma} \|Y - V^\nu\|^2_{\mathcal{E}} \right\} \\ V^{\nu+1} &= \psi Y^{\nu+1} \\ \gamma Z^{\nu+1} &= \gamma U^\nu + X^\nu + Y^{\nu+1} \\ U^{\nu+1} &= U^\nu + \frac{1}{\gamma}[Y^{\nu+1} - V^{\nu+1}] \end{aligned}$$

Eliminating intermediate variables and decomposing the first equality into scenario subproblems we obtain the progressive hedging algorithm, with the notations $Y = X$, $V = \hat{X}$ and $U = W$. That is

$$\begin{aligned} X^{\nu+1}(s) &= \arg\min_{y \in C(s)} \left\{ f(y) + \langle W^\nu(s), y - V^\nu(s) \rangle + \frac{1}{2\gamma} \|y - V^\nu(s)\|^2 \right\}, \forall s \in S \\ \hat{X}_t^{\nu+1}(s) &= \sum_{\omega \in A} p_\omega U_t^\nu(\omega) / \sum_{\omega \in A} p_\omega, \forall s \in A \in \mathcal{A}_t, \forall t \\ W^{\nu+1} &= W^\nu + \frac{1}{\gamma}(X^{\nu+1} - \hat{X}^{\nu+1}) \end{aligned}$$

4.5 Convergence Analysis

In the last section it was shown that the progressive hedging algorithm can be derived from the Douglas Rachford splitting method or from the Spingarn's partial inverse method which are both applications of the proximal point algorithm.

Hence, the convergence properties are inherited from this theory. However, in this section we will show convergence results that work in connection with the augmented Lagrangian by presenting the algorithm as a proximal step of a certain saddle function relative to the “variance” norm.

We will show the form of the algorithm in which the exact solutions for the subproblems are calculated in Step 1. This is known as the **exact minimization** case.

Theorem 4.3 *Let* $\{\hat{X}^\nu\}_{\nu=1}^\infty$ *and* $\{W^\nu\}_{\nu=1}^\infty$ *be the sequences generated by the progressive-hedging algorithm. These sequences will be bounded if and only if optimal solutions exist for the problem* $(\mathcal{P})$ *and its dual problem*$(\mathcal{D})$ *given by*

$$\max_{W \in \mathcal{M} \cap D} \inf_{X \in \mathcal{C}} L(X, W), \text{ where} D = \{W : \inf_{X \in \mathcal{C}} L(X, W) > -\infty\}. \qquad (\mathcal{D})$$

In this case, for some particular X^* *and* W^* *optimal solutions of* $(\mathcal{P})$ *and* $(\mathcal{D})$ *respectively it will be true that*

$$\hat{X}^\nu \to X^* \text{ and } W^\nu \to W^* \tag{4.16}$$

Furthemore, in terms of the following norm

$$\|(X, Y)\|_r = (\|X\|_\mathcal{E}^2 + r^{-2}\|Y\|_\mathcal{E}^2)^{1/2} \tag{4.17}$$

one will have in every iteration $\nu = 0, 1, 2, \ldots$ *that*

$$\|(\hat{X}^{\nu+1}, W^{\nu+1}) - (X^*, W^*)\|_r \leq \|(\hat{X}^\nu, W^\nu) - (X^*, W^*)\|_r \tag{4.18}$$

with strict inequality unless $(\hat{X}^\nu, W^\nu) = (X^*, W^*)$. *On will also have in every iteration that*

$$\|(\hat{X}^{\nu+1}, W^{\nu+1}) - (\hat{X}^\nu, W^\nu)\|_r \leq \|(\hat{X}^\nu, W^\nu) - (\hat{X}^{\nu-1}, W^{\nu-1})\|_r \tag{4.19}$$

Proof Consider the following rescaling of the multiplier vectors

$$\overline{W} = r^{-1}W, \quad \overline{W}^\nu = r^{-1}W^\nu, \quad \overline{W}^* = r^{-1}W^*$$

then

$$\|(X, W)\|_r = (\|X\|_\mathcal{E}^2 + \|\overline{W}\|_\mathcal{E}^2)^{1/2}$$

Consider now the **projected saddle function**

$$l_r(V, \overline{W}) = \inf\{r^{-1}F(X) + \langle X, \overline{W}\rangle : X \in \mathcal{C}, \psi X = V\} \text{ for } V \in \mathcal{N}, \overline{W} \in \mathcal{M} \tag{4.20}$$

We shall show that the progressive hedging algorithm in the convex case with exact minimization takes the form that $(\hat{X}^{\nu+1}, \overline{W}^{\nu+1})$ is the unique saddle point of

$$l_r^\nu(V, \overline{W}) = l_r(V, \overline{W}) + \frac{1}{2}\|V - \hat{X}^\nu\| - \frac{1}{2}\|\overline{W} - \overline{W}^\nu\|^2 \tag{4.21}$$

Associated with l_r is the multivalued function $T_r : \mathcal{N} \times \mathcal{M} \to 2^{\mathcal{N} \times \mathcal{M}}$ defined by

$$T_r(V, \overline{W}) = \{(Y, -U) : (Y, U) \in \partial l_r(V, \overline{W})\}$$

or equivalent

$$T_r(V, \overline{W}) = \{(Y, -U) : (Y, -\overline{W}) \in \partial \psi_r(V, U)\}$$

This operator is maximal monotone. Our claim that $(\hat{X}^{\nu+1}, \overline{W}^{\nu+1})$ is the unique saddle point of l_r^ν means that

$$(0,0) \in \partial l_r^\nu(\hat{X}^{\nu+1}, \overline{W}^{\nu+1}) = \partial l_r(\hat{X}^{\nu+1}, \overline{W}^{\nu+1}) + (\hat{X}^{\nu+1} - \hat{X}^\nu, \overline{W}^\nu - \overline{W}^{\nu+1})$$

or in terms of T_r that

$$(\hat{X}^\nu - \hat{X}^{\nu+1}, \overline{W}^\nu - \overline{W}^{\nu+1}) \in T_r(\hat{X}^{\nu+1}, \overline{W}\nu + 1). \tag{4.22}$$

In the original context of F and $\mathcal{C}$ We have

$$(Y, -\overline{W}) \in \partial\varphi_r(V, U) \Leftrightarrow Y - \overline{W} \in \partial(r^{-1}F + I_\mathcal{C})(X), \text{ where } X = V + U$$

Moreover the subdifferentiation rules of convex analysis yield

$$\partial(r^{-1}F + I_\mathcal{C})(X) = r^{-1}\partial F(X) + N_\mathcal{C}(X)$$

Therefore

$$(Y, -U) \in T_r(V, \overline{W}) \Leftrightarrow Y - \overline{W} \in r^{-1}\partial F(X) + N_\mathcal{C}(X), \text{ where} \tag{4.23}$$
$$X = V + U, V \in \mathcal{N}, U \in \mathcal{M}, Y \in \mathcal{N}. \tag{4.24}$$

We can now transmute our claim (4.22) into the equivalent form

$$\hat{X}^\nu - \hat{X}^{\nu+1} - \overline{W}^{\nu+1} \in r^{-1}\partial F(\hat{X}^{\nu+1} + \overline{W}^{\nu+1} - \overline{W}^\nu) + N_\mathcal{C}(\hat{X}^{\nu+1} + \overline{W}^{\nu+1} - \overline{W}^\nu). \tag{4.25}$$

From the updating formula in the progressive hedging

$$W^{\nu+1} = W^\nu + r[X^{\nu+1} - \hat{X}^{\nu+1}]$$

we get $\overline{W}^{\nu+1} - \overline{W}^{\nu} = X^{\nu+1} - \hat{X}^{\nu+1}$ and then

$$\hat{X}^{\nu+1} + \overline{W}^{\nu+1} - \overline{W}^{\nu} = X^{\nu+1},$$

$$\hat{X}^{\nu} - \hat{X}^{\nu+1} - \overline{W}^{\nu+1} = \hat{X}^{\nu} - X^{\nu+1} - r^{-1}W^{\nu}$$

Thus our claim (4.25) is the same as

$$0 \in \partial F(X^{\nu} + 1) + N_{\mathcal{C}}(X^{\nu+1}) + W^{\nu} + r(X^{\nu+1} - \hat{X}^{\nu})$$

or better still:

$$0 \in \partial F^{\nu}(X^{\nu+1}) + N_{\mathcal{C}}(X^{\nu+1}) \tag{4.26}$$

where F^{ν} is the objective in the subproblem $(\mathcal{P}^{\nu})$ in Step 1 of the progressive hedging algorithm

$$F^{\nu}(X) = F(X) + \langle X, W^{\nu}\rangle + \frac{1}{2}r\|X - \hat{X}^{\nu}\|^2 \tag{4.27}$$

But we are working in a framework of convex programming, so the subdifferential condition (4.26) is both necesary and sufficient for

$$X^{\nu+1} \in \arg\min_{X\in\mathcal{C}} F^{\nu}(X) = \operatorname{argmin}(\mathcal{P}^{\nu}) \tag{4.28}$$

The uniqueness mentioned in (4.21) is evident from the stric convexity-concavity induced on the function l_r^{ν} by the proximal terms in V and $\overline{W}$. It is equivalent also to the uniqueness of $X^{\nu+1}$ in (4.28), which comes from the proximal term in (4.27) . The expression (4.22) can be writing as

$$(\hat{X}^{\nu+1}, \overline{W}^{\nu+1}) = M_r(\hat{X}^{\nu}, \overline{W}^{\nu}), \tag{4.29}$$

where $M_r = (I + T_r)^{-1} = J_{T_r}$ which is firmly nonexpansive because is the resolvent of a maximal monotone operator. This means in the notation

$$Z = (V, \overline{W}), \quad , Z^{\nu} = (\hat{X}^{\nu}, \overline{W}^{\nu}), \quad Z^{*} = (V^{*}, \overline{W}^{*}) \tag{4.30}$$

that one has

$$\|M_r(Z') - M_r(Z)\|_r^2 + \|(Z' - M_r(Z))\|^2 \leq \|Z' - Z\|^2. \tag{4.31}$$

From equation (4.29) one has the proximal point algorithm associated with the maximal monotone operator T_r. So the sequence $\{(\hat{X}^{\nu}, \overline{W}^{\nu})\}_{\nu}$ is bounded if and only if there is a pair $(V^{*}, \overline{W}^{*})$ such that

$$(0,0) \in T_r(V^{*}, \overline{W}^{*}), \text{ with } V^{*} \in \mathcal{N}, \overline{W}^{*} \in \mathcal{M} \tag{4.32}$$

in wich case the sequence actually converges to some such pair. Due to (4.23), the pair $(V^*, \overline{W}^*)$ satisfies

$$-\overline{W}^* \in r^{-1}\partial F(X^*) + N_{\mathcal{C}}(X^*) \textit{ for } X^* = V^*$$

this relation can also be written as

$$-W^* \in \partial F(X^*) + N_{\mathcal{C}}(X^*) \textit{ for } X^* = V^* \tag{4.33}$$

then X^* being optimal for $(\mathcal{P})$ and W^* being an associated multiplier that solves the dual problem. It only remains to check that the iterations of the progressive hedgin are improving with each iteration. In the notation (4.30) we have to prove that

$$\|Z^{\nu+1} - Z^*\|_r \leq \|Z^\nu - Z^*\| \text{ for all } \nu \tag{4.34}$$

with strict inequality unless $Z^\nu = Z^*$ Noting that the optimality relation (4.32) can be written as $Z^* = M_r(Z^*)$, whwereas $Z^{\nu+1} = M_r(Z^\nu)$ for every ν we can get then

$$\|Z^{\nu+1} - Z^*\|_r^2 + \|(Z^\nu - Z^{\nu+1} - 0)\|_r \leq \|Z^\nu - Z^*\|_r^2$$

Finally

$$\|Z^{\nu+1} - Z^*\|_r \leq \|Z^\nu - Z^*\|_r \text{ for all } \nu \tag{4.35}$$

□

Therefore, the exact implementation of the algorithm leaves us with policies that at each iteration are better than the previous ones. Thus, we can decide to stop at a certain iteration if convergence is slow.

Chapter 5

Applications

We will discuss as a first application a simplified version of the multistage hydrothermal dispatch problem presented in [13] without intervention of thermal energy. This problem has the following specifications for a period of time $\{0, \ldots, T-1\}$.

1. The hydroelectric system consists of a single hydroelectric with N independent reservoirs (not cascaded). The function of the reservoir $n \in \{1, \ldots, N\}$ is to store and discharge water for the production of electricity.

2. For each reservoir n, the water flow input in stage t is a stochastic parameter which only takes real values on the set $\Omega = \{\xi_{dry}, \xi_{wets}\}$ with respective probabilities $p(\xi_{dry})$ and $p(\xi_{wets}) = 1 - p(\xi_{dry})$. Furthermore, the water flow output in stage t are given by the turbinate volume (this volume is used to generate energy) and the discharged volume. The diagram of a reservoir in this case is shown in the Figure 5.3.

3. We will assume that for each reservoir n the volume of the initial stage ($t = 0$) is known.

4. The objective is at least to supply the energy demand D_t in each stage t throughout the planning period with the minimum cost. In case the demand is not satisfied with the hydroelectric energy, it will be considered to buy energy from an external company. The demand D_t will be a fixed parameter at each stage t.

Modeling the problem. The set of scenarios S is determined only by the water flow input, hence

$$S = \underbrace{\Omega \times \ldots \times \Omega}_{T \text{ times}}$$

A particular scenario $s \in S$ is given by $s = \{a_1^s, a_2^s, \ldots, a_T^s\}$, where $a_t^s \in \Omega$ indicates the inflow of water observed in scenario s at the stage t. The scenario subproblem $(\mathcal{P}_s)$ then consists of determining for each stage t and reservoir n,

- $y_{t,n}(s)$: Quantity of water to convert to energy for each reservoir n (turbinate water),
- $r_{t,n}(s)$: Quantity of water to discharge from the reservoir n,
- $\theta_t(s)$: Energy to buy from an external company at stage t.

Subject to the following constraints: For each stage $t \in \{0, \ldots, T-1\}$ and reservoir $n \in \{1, \ldots, N\}$

- Water balance equation:

$$v_{t+1,n}(s) = v_{t,n}(s) + a_t^s - y_{t,n}(s) - r_{t,n}(s) \tag{5.1}$$

  where $v_{t,n}(s)$ and $v_{t+1,n}(s)$ are respectively the initial and final volume of the reservoir n in the scenario s.

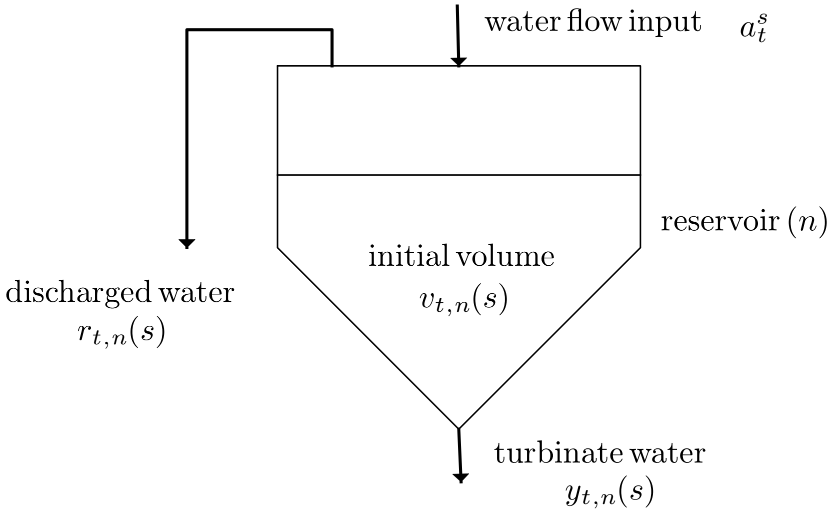

Figure 5.1: The diagram of reservoir n shows the water balance equation for stage t, given a scenario s

- Volume restrictions:

$$\begin{aligned}0 &\le v_{t,n}(s) \le \bar{v}_n\\ 0 &\le y_{t,n}(s) \le \bar{y}_n\\ 0 &\le r_{t,n}(s)\end{aligned}$$

  where the constant $\bar{v}_n$ is the maximum volume of reservoir n and $\bar{y}_n$ is the maximum volume that can be turbinated.

- The demand equation

$$\sum_{n=1}^{N} \rho_n y_{t,n}(s) + \theta_t(s) \ge D_t \tag{5.2}$$

  where ρ_n is the constant power of the reservoir n which multiplied by the amount of water $y_{t,n}(s)$, gives us the energy generated by reservoir n.

We denote by $X_t(s) = (y_t(s), r_t(s), \theta_t(s))$ the decision vector in stage t given the scenario s. Where

$$y_t(s) = (y_{t,1}(s), \dots, y_{t,N}(s)) \text{ and } r_t(s) = (r_{t,1}(s), \dots, r_{t,N}(s)).$$

It is clear that $X_t(s) \in \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}$. All these vectors will be components of the scenario subproblem decision $X(s) = (X_0(s), \dots, X_{T-1}(s))$.

Finally, we get the following scenario subproblem

$$\min_{X(s)\in C_s} \left[\sum_{t=0}^{T-1} \langle c, y_t(s)\rangle + \sum_{t=0}^{T-1} \delta\theta_t(s)\right] \tag{$\mathcal{P}_s$}$$

where $c \in \mathbb{R}^N$ is the cost vector whose components are the unit costs of energy production of the reservoir n. The constant $\delta \in \mathbb{R}$ is the penalty cost for not supply the demand and C_s is defined by

$$C_s = \left\{\begin{array}{rl} \sum_{n=1}^{N} \rho_n y_{t,n}(s) + \theta_t(s) \ge D_t & \forall 0 \le t \le T-1\\ v_{t+1,n}(s) = v_{t,n}(s) + a_t^s - y_{t,n}(s) - r_{t,n}(s) & \forall 1 \le t \le T-1, 1 \le n \le N\\ v_{t+1,n}(s) = V_{0,n} + a_t^s - y_{t,n}(s) - r_{t,n}(s) & t = 0, 1 \le n \le N\\ 0 \le v_{t,n}(s) \le \bar{v}_n & \forall 0 \le t \le T-1, 1 \le n \le N\\ 0 \le y_{t,n}(s) \le \bar{y}_n & \forall 0 \le t \le T-1, 1 \le n \le N\\ 0 \le r_{t,n}(s) & \forall 0 \le t \le T-1, 1 \le n \le N \end{array}\right.$$

With $V_{0,n}$ as the initial volume of reservoir n wich is known. To formulate the set of nonanticipative policies $\mathbb{N}$, let's define $\mathcal{A} = \{\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_{T-1}\}$ the set of partitions in each stage t. Since S has 2^T elements then

$$\mathcal{A}_t = \{\{s_1, \dots, s_{2^T/2^{t+1}}\}, \dots, \{s_{2^T - 2^T/2^{t+1}+1}, \dots, s_{2^T}\}\} \tag{5.3}$$

Then a policy $X : S \to \mathbb{R}^b$ is nonanticipative $(\in \mathcal{N})$ with $b = (2N+1)T$ if and only if for all $t = \{0, \dots, T-1\}$ holds

$$x_t(s) = x_t(s'), \forall s, s' \in A \in \mathcal{A}_t \Leftrightarrow \forall s, s' \in A \in \mathcal{A}_t \left\{ \begin{array}{rcl} y_t(s) & = & y_t(s') \\ r_t(s) & = & r_t(s') \\ \theta_t(s) & = & \theta(s') \end{array} \right.$$

5.1 Multistage Problem

For the computational implementation we generate a problem with the following data: $T = 4$ time periods, the demand on each period is shown in the next table

t	0	1	2	3
D_t (MW)	1800	2300	1600	1450

We also consider $N = 4$ reservoirs with their respective volume data, power constants and unit cost of production

n	1	2	3	4
$\bar{y}_n$(L)	500	480	610	450
$\bar{v}_n$(L)	7000	7900	9000	9850
$V_{0,n}$(L)	200	180	175	210
ρ_n(MW/L)	1.2	2.1	1.9	2.5
c_n(\$/MW)	100	120	90	135

The flow values of water are two, $\xi_{dry} = 63$ and $\xi_{wets} = 220$ with respective probabilites $p(\xi_{dry}) = 0.6$ and $p(\xi_{wets}) = 0.4$. We also assume that unit external energy is purchased at a fixed cost of 1000 ($/MW).

Since $T = 4$ we have a total of 16 scenarios. We solve each unmodified scenario subproblem $(\mathcal{P}_s)$ which is a linear programming problem using the *scipy.optimize.linprog* tool from Python language. Thus obtaining an initial policy X^0 (shown in appendix) for the Progressive Hedging Algorithm. The modified scenario subproblem $(\mathcal{P}_s^\nu)$ is a quadratic optimization problem that was solved using the *quadprog* solver to compute the optimal solution with high precision. We obtained for 8 iterations the following values of $\|KX\|^2$

iteration	$\|KX\|^2$
1	6.754618345511294e-26
2	6.212856696144855e-26
3	3.2451598926544576e-26
4	7.676695609979558e-27
5	4.3103793242524754e-27
6	2.8159251069319285e-27
7	3.291808818020291e-28
8	2.462427157828823e-28

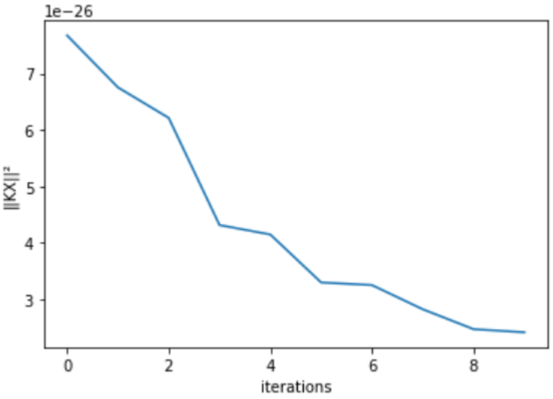

Figure 5.2: $\|KX\|^2$ value variation. The strict decreasing can be observed approximately, which is corroborated by Theorem 4.3

5.2 Two Stage Problem

In this section, two-stage problem test is presented to compare the methods shown and their efficiency over conventional methods. These problems will be posed in a hydrothermal dispatch framework to illustrate the importance of considering the randomness of the flows in the operation of reservoirs.

We generate a problem with the following data: $T = 2$ time periods, the demand on each period is shown in the next table

t	0	1
D_t (MW)	50	67

We also consider $N = 2$ reservoirs with their respective volume data, power constants and unit cost of production

n	1	2
$\bar{y}_n$(L)	10	12
$\bar{v}_n$(L)	60	50
$V_{0,n}$(L)	17	19
ρ_n(MW/L)	2.8	3.8
c_n(\$/MW)	41	15

The volume discharged will be penalized for the first reservoir with 243\$/L and for the second reservoir with 308\$/L and penalty of 1033\$/MW.

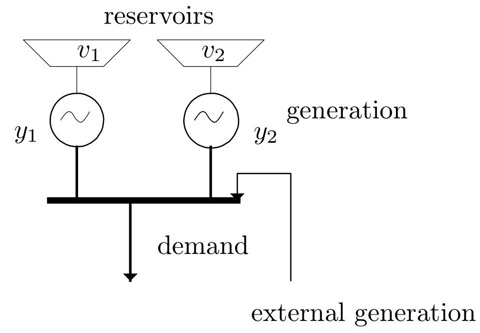

Figure 5.3: topological diagram of the problem

5.2.1 Progressive Hedging

As established in the previous section, the optimization problem with explicit non-anticipative constraints is

$$
\begin{array}{ll}
\min & \sum_{k=1}^{3} p_k \left[\sum_{t=1}^{2} (41 y_{t,1}^k + 15 y_{t,2}^k + 243 s_{t,1}^k + 308 s_{t,2}^k + 1033 \alpha_t^k) \right] \\
s.t & 2.8 y_{1,1}^k + 3.8 y_{1,2}^k + \alpha_1^k \geq 50 \\
 & 2.8 y_{2,1}^k + 3.8 y_{2,2}^k + \alpha_2^k \geq 67 \\
 & v_{1,1}^k = 57 - y_{1,1}^k - s_{1,1}^k, \forall k = 1,2,3 \\
 & v_{2,1}^k = v_{1,1} + \xi_1^k - y_{2,1} - s_{2,1} \\
 & v_{1,2}^k = 49 - y_{1,2}^k - s_{1,2}^k \\
 & v_{2,2}^k = v_{1,2} + \xi_2^k - y_{2,2} - s_{2,2} \\
 & 0 \leq v_{t,1}^k \leq 60, 0 \leq y_{t,1}^k \leq 10 \\
 & 0 \leq v_{t,2}^k \leq 50, 0 \leq y_{t,2}^k \leq 12 \\
 & 0 \leq s_{t,i}, \forall i = 1,2 \\
 & y_{1,1}^k = \sum_{j=1}^{3} p_j y_{1,1}^j \\
 & y_{1,2}^k = \sum_{j=1}^{3} p_j y_{1,2}^j . \forall ; k = 1,2,3; t = 1,2.
\end{array}
\tag{5.4}
$$

where (ξ_1^k, ξ_2^k) are the inflows at stage 2 in scenario k with probability p_k for the reservoir 1 and 2 respectively. The nonanticipativity constraint are

$$
y_{1,1}^k = \sum_{j=1}^{3} p_j y_{1,1}^j \tag{5.5}
$$

$$
y_{1,2}^k = \sum_{j=1}^{3} p_j y_{1,2}^j . \tag{5.6}
$$

Let us remember that these equations indicate that in our first stage we made a decision that does not depend on the scenarios. We solve this problem using Python 3.0 language[1]. Scenario subproblems are quadratic optimization problems with parameter $r = 40$ and were solved with *qpsolver* package.

The following two initial policies were tested:

Optimal Initial Policy

Looking at the problem (5.4) without the nonanticipativity constratints, we find that it is separable into scenario subproblems. Theoretically (Chapter 3), it is

[1] The code of the implementation of these experiments will be in https://github.com/dvid1999

the best starting policy for progressive Hedging Algorithm :

$$\hat{x}^1 = (9.50, 10.6, 2.07, 1.79, 0.00, 10.0, 12.0, 159, 2.02, 3.00)$$
$$\hat{x}^2 = (9.50, 10.6, 2.07, 1.79, 0.00, 10.0, 12.0, 62.63, 0.00, 3.00)$$
$$\hat{x}^3 = (9.50, 10.6, 2.07, 1.79, 0.00, 10.0, 12.0, 0.00, 0.00, 3.00)$$

Hence, the following values of $||K\boldsymbol{x}||^2$ where obtained:

it	$\|\|K\boldsymbol{x}\|\|^2$	it	$\|\|K\boldsymbol{x}\|\|^2$
1	1.109	11	0.008
2	0.195	12	0.003
3	0.094	13	0.004
4	0.059	14	0.002
5	0.052	15	0.002
6	0.029	16	0.001
7	0.026	17	0.001
8	0.015	18	0.00
9	0.016		
10	0.008		

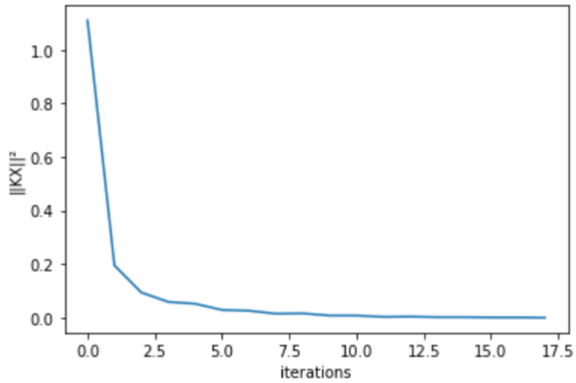

Figure 5.4: Values of $||K\boldsymbol{x}||^2$ obtained by the optimal initial policy, they are measuring how far one is from satisfying all the constraints

The optimal policy is

$$x^1 = (10, 11.58, 0.07, 0.0, 0.00, 10.0, 12.0, 159, 2.02, 3.00)$$
$$x^2 = (10, 11.58, 0.07, 0.0, 0.00, 10.0, 12.0, 62.63, 0.00, 3.00)$$
$$x^3 = (10, 11.58, 0.07, 0.0, 0.00, 10.0, 12.0, 0.07, 0.00, 3.00)$$

Zero Initial Policy

We can actually start from any policy in the nonanticipative subspace.A particular choose is the zero policy, that is:

$$
\begin{aligned}
x^1 &= (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)\\
x^2 &= (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)\\
x^3 &= (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)
\end{aligned}
$$

The following values of $||K\boldsymbol{x}||^2$ where obtained:

it	$\|\|K\boldsymbol{x}\|\|^2$	it	$\|\|K\boldsymbol{x}\|\|^2$	it	$\|\|K\boldsymbol{x}\|\|^2$
1	197.997	10	0.481	19	0.012
2	20.108	11	0.323	20	0.009
3	11.113	12	0.259	21	0.004
4	1.254	13	0.167	22	0.004
5	2.824	14	0.135	23	0.002
6	2.019	15	0.076	24	0.002
7	1.312	16	0.063	25	0.001
8	0.93	17	0.026	26	0.001
9	0.612	18	0.022	27	0.0

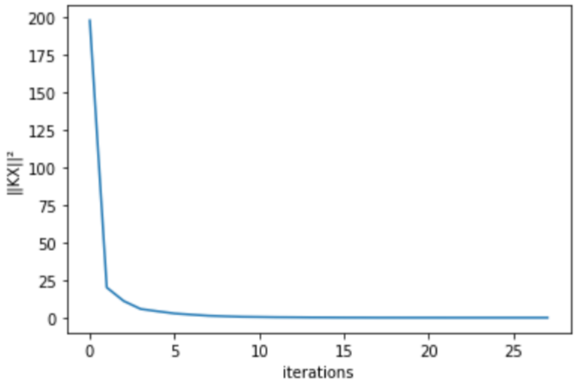

Figure 5.5: Values of $||K\boldsymbol{x}||^2$ obtained by the zero initial policy, they are measuring how far one is from satisfying all the constraints

The optimal policy is (by the approximation of the decimals)

$$
\begin{aligned}
x^1 &= (10, 11.47, 0.58, 0.0, 0.00, 10.0, 12.0, 161.44, 2.02, 3.00)\\
x^2 &= (10, 11, 47, 0.58, 0.0, 0.00, 10.0, 12.0, 61.44, 0.58, 3.00)\\
x^3 &= (10, 11.47, 0.58, 0.0, 0.00, 10.0, 12.0, 24.58, 0.00, 5.82)
\end{aligned}
$$

5.2.2 Parameter r

For the same problem, several values of r were used. Each giving us different iteration times. It was observed for the problem that for small r values it took longer for the $||KX||^2$ norm to reach zero.

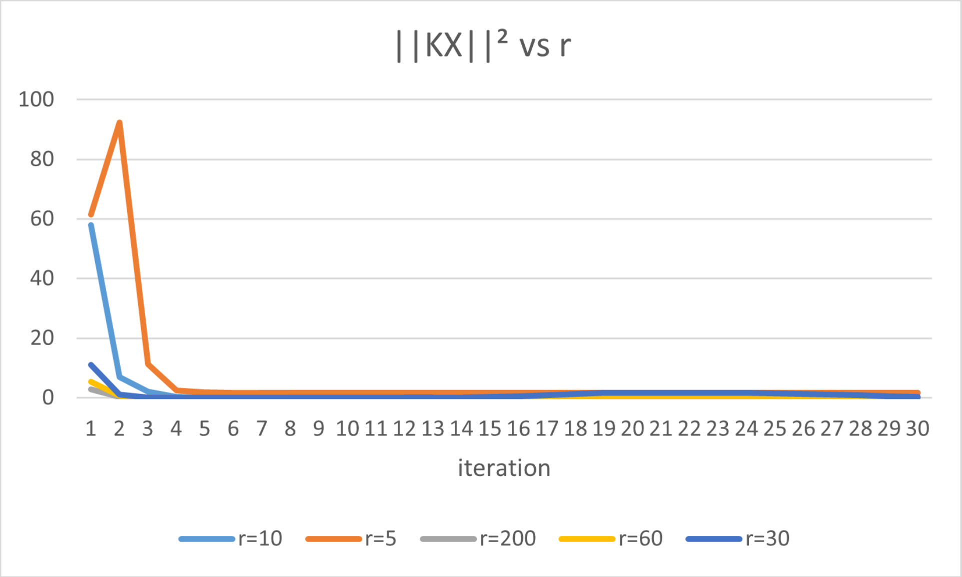

Figure 5.6: Behavior of the variance for different values of r chosen.

5.2.3 L-shaped Method

The nested formulation of the problem allow us form the following problem:

$$\begin{aligned}
\min \quad & 41y_{1,1} + 15y_{1,2} + 243s_{1,1} + 308s_{1,2} + 1033\alpha_1 + \\
& \sum_{k=1}^{3} p_k \left[(41y_{2,1}^k + 15y_{2,2}^k + 243s_{2,1}^k + 308s_{2,2}^k + 1033\alpha_2^k) \right] \\
s.t \quad & 2.8y_{1,1} + 3.8y_{1,2} + \alpha_1 \geq 50 \\
& 2.8y_{2,1}^k + 3.8y_{2,2}^k + \alpha_2^k \geq 67 \\
& v_{1,1} = 57 - y_{1,1} - s_{1,1} \\
& v_{2,1}^k = v_{1,1} + \xi_1^k - y_{2,1}^k - s_{2,1}^k \\
& v_{1,2} = 49 - y_{1,2} - s_{1,2} \\
& v_{2,2}^k = v_{1,2} + \xi_2^k - y_{2,2}^k - s_{2,2}^k \\
& 0 \leq v_{1,1}, v_{2,1}^k \leq 60, 0 \leq y_{1,1}, y_{2,1}^k \leq 10 \\
& 0 \leq v_{1,2}, v_{2,2}^k \leq 50, 0 \leq y_{1,2}, y_{2,2}^k \leq 12 \\
& 0 \leq s_{1,i}, s_{2,i}^k, \forall i = 1, 2; k = 1, 2, 3.
\end{aligned} \tag{5.7}$$

Note that unlike the explicit formulation of non-anticipation, the first-stage decision vector implicitly indicates that it does not depend on the scenarios. The initial master problem is the first-stage problem without cuts, that is :

$$\begin{aligned} \min \quad & 41y_{1,1} + 15y_{1,2} + 243s_{1,1} + 308s_{1,2} + 1033\alpha_1 \\ s.t \quad & 2.8y_{1,1} + 3.8y_{1,2} + \alpha_1 \geq 50 \\ & v_{1,1} = 57 - y_{1,1} - s_{1,1} \\ & v_{1,2} = 49 - y_{1,2} - s_{1,2} \\ & 0 \leq v_{1,1}, \leq 60, 0 \leq y_{1,1} \leq 10 \\ & 0 \leq v_{1,2} \leq 50, 0 \leq y_{1,2} \leq 12 \\ & 0 \leq s_{1,1}, s_{1,2} \end{aligned} \tag{5.8}$$

The optimal solution to this problem is:

$$y^*_{1,1} = 5, y^*_{1,2} = 12, s^*_{1,1} = 0, s^*_{1,2} = 0, \alpha^*_1 = 0 \tag{5.9}$$

This vector is introduced as initial information in the L-shaped method obtaining in three iterations

$y_{1,1}$	$y_{1,2}$	$s_{1,1}$	$s_{1,2}$	α_1
10	12	0	0	0

Table 5.1: First-stage optimal decision

The variation of the cost function in the L-shaped is as follows:

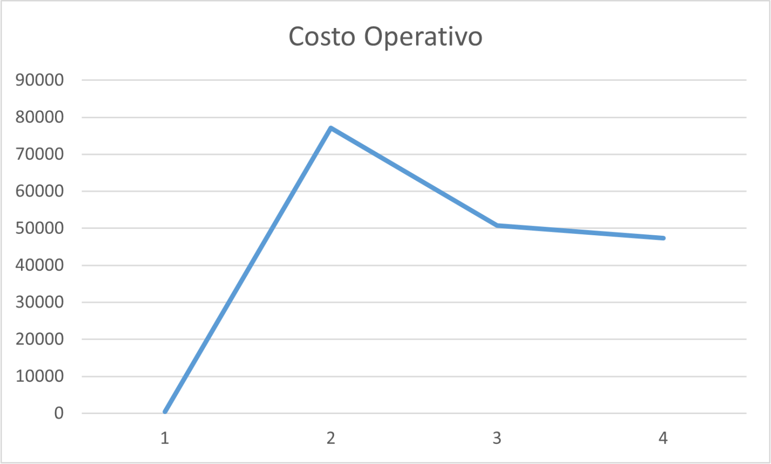

Figure 5.7: The first value corresponds only to the first stage, while the other values correspond to the total cost of the first and second stage, arriving at the approximate optimum value of $47369.

Chapter 6

Conclusions

1. Two-stage optimization problems were solved with the L-shaped algorithm and the progressive-hedging algorithm.Although these problems have few levels, they have a large dimension and this tends to grow if more scenarios are considered. It has been observed that in a few iterations we can reach a considerable decision using the studied algorithms. This is not the case if we solve the problem with conventional linear programming methods. Thus, decomposition algorithms strike a balance between solution accuracy and computational time.

2. Compared to the L-shaped method, the progressive-hedging algorithm depends on a parameter r, which can be adjusted to obtain different convergence times. However, the progressive-hedging algorithm also solves a modified scenario subproblem separately, which is sometimes considered useful information.

3. As established in section 4.1.1 Projected problem, the progressive-hedging algorithm attempts to trace a path from $\hat{X}^0$ to the optimal policy $\hat{X}^*$. So we usually start by solving each scenario subproblem separately (as long as it is easy to solve) to obtain an even faster convergence time. This does not mean that it is always convenient to do this, so we can also start from the null policy.

Bibliography

[1] R. T. Rockafellar and R. J-B Wets. “Scenarios and policy aggregation in optimization under uncertainty”. Mathematics of operations research **16** (1991), 119–147.

[2] G. Bareilles, Y. Laguel, D. Grishchenko, F. Iutzeler and J. Malick. “Randomized Progressive Hedging methods for Multi-stage Stochastic Programming”. Annals of Operation Research, Springer Verlag, In press **295** (2020).

[3] H.H. Bauschke and P.L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces* (2 ed.). Springer Science & Business Media (2017).

[4] J. Sun, H. Xu and M. Zhang, “A new interpretation of the Progressive Hedging algorithm for multistage stochastci minimization problems”. Journal of Industrial and Management Optimization **16** (2020), 1655-1662.

[5] R.T. Rockafellar. *Optimization under uncertainty, Lecture notes*. University of Washington, 2001.

[6] R. J-B Wets. “The Aggregation Principle in Scenario Analysis and Stochastic Optimization”. Algorithms and Modelling in Mathematical Programming ed. S. Wallace, Springer-Verlag, Berlin (1989), 91-113.

[7] A. Ruszczyński, and A. Shapiro . *Stochastic programming models*. Handbooks in operations research and management science **10** (2003), 1-64.

[8] J.E. Spingarn. “Partial inverse of a monotone operator”. Applied Mathematics and Optimization **10** (1983),247–265.

[9] R.T. Rockafellar. *Convex Analyis*. Princeton University Press, 1970.

[10] A. Shapiro, D. Dentcheva and A. Ruszczyński: *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.

[11] EPRI, Electric generation expansion analysis system (EGEAS), EPRI Report EL2561, Aug. 1982.

[12] P.B.D.Massé, Les réserves et la régulation de l'avenir dans la vie économique, Hemann, Paris, 1946.

[13] N.M.Campodonico. "Representación analítica de fallas de los equipamientos y variación de la demanda en el despacho hidrotérmico multietapa". PhD thesis, Universidade Federal do Rio de Janeiro, COPPE (1997).

[14] J.P.C. Kleijnen, Statistical Techniques in simulation, Marcel Dekker, 1974.

[15] R.R.Booth, Power system simulation model based on probability analysis IEEE, PICA Conference, 1971.

[16] John R. Birge and Francois Louveaux. *Introduction to Stochastic Programing*. Springer New York 2011.

[17] Martin Biel and Mikael Johansson. Distributed L-shaped algorithms in Julia.

[18] R.T Rockafellar. Monotone operators and the proximal point algorithm *SIAM Journal on Control and Optimization*, 14(5):877-898,1976.