UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA

FILIPE GOULART CABRAL

THE ROLE OF EXTREME POINTS FOR CONVEX HULL OPERATIONS

Rio de Janeiro
2018

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA

FILIPE GOULART CABRAL

THE ROLE OF EXTREME POINTS FOR CONVEX HULL OPERATIONS

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Matemática, Instituto de Matemática da Universidade Federal do Rio de Janeiro (UFRJ), como parte dos requisitos necessários à obtenção do título de Mestre em Matemática.

Advisor: Bernardo Freitas Paulo da Costa

Rio de Janeiro
2018

CBIB Cabral, Filipe Goulart

The role of extreme points for convex hull operations / Filipe Goulart Cabral. – 2018.
172 f.: il.

Dissertação (Mestrado em Matemática) – Universidade Federal do Rio de Janeiro, Instituto de Matemática, Programa de Pós-Graduação em Matemática, Rio de Janeiro, 2018.

Advisor: Bernardo Freitas Paulo da Costa.
.

1. Programação Inteira Mista Estocástica. 2. Programação Inteira Mista. 3. Programação Disjuntiva. 4. Análise Convexa. – Teses. I. da Costa, Bernardo Freitas Paulo (Orient.). II. Universidade Federal do Rio de Janeiro, Instituto de Matemática, Programa de Pós-Graduação em Matemática. III. Título

CDD

FILIPE GOULART CABRAL

The role of extreme points for convex hull operations

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Matemática, Instituto de Matemática da Universidade Federal do Rio de Janeiro (UFRJ), como parte dos requisitos necessários à obtenção do título de Mestre em Matemática.

Aprovado em: Rio de Janeiro, _____ de ______________ de _________.

Prof. Bernardo Freitas Paulo da Costa (Orientador)

Prof. Alexandre Street de Aguiar

Dr. Mario Veiga Ferraz Pereira

Prof. Nelson Maculan Filho

Prof. Shabbir Ahmed

*To all Brazilians who dream
of a fair and prosperous country.*

ACKNOWLEDGEMENTS

I would first like to thank my mother Rita, sister Daniella and family for their ongoing support and encouragement during all my academic endeavors. My heartfelt thanks go to my girlfriend Liliane Portugal for her love and affection that contributed expressively to the success of this dissertation.

I am profoundly grateful to my advisor Bernardo Costa for all his partnership and dedication that allowed me to go beyond my own expectations. It worth emphasizing that my research would have been impossible without the aid and support of Joari Costa, whose guidance, comments and teachings go beyond the scope of this work. I would also like to extend my deepest gratitude to Evandro Mendes for his encouragement and advice since I started working at ONS and for his remarkable motto to hard situations: "discernment, calmness and perseverance".

My sincere thanks to professors Shabbir Ahmed, Mario Veiga, Nelson Maculan and Alexander Street for being my dissertation examiners and providing valuable comments for this dissertation. I am also gratefully indebted to professor Alexander Shapiro for his helpful advice to this work and all his teaching since I joined ONS.

A very special thank you to Alberto Kligerman, Mario Daher and Francisco Arteiro from ONS for believing in my research and for supporting the technical cooperation agreement between ONS and UFRJ that resulted in this master dissertation.

I cannot forget mentioning friends from UFRJ who went through hard times together, cheered me on, and celebrated each accomplishment: Claudio Verdun, Ivani Ivanova, Guilherme Sales, Leonardo Assumpção, Vitor Luiz, Iago Leal, Hugo Carvalho, and professors Fábio Ramos, Marcelo Tavares and Bruno Scardua from the Mathematics Department, Carolina Effio and professor José Herskovits from the Mechanical Engineering Departartment, professor Abilio Lucena from the Computational Engineering Department, and professor Suely Freitas from the Chemical Engineering Department.

Some special words of gratitude go to my friends from ONS that have always been a major source of support when things would get a bit discouraging: Paulo Nascimento, Lucas Khenayfis, Alessandra Mattos, Francislene Madeira, Candida Abib, Maria Helena Azevedo, Liciane Pataca, Hugo Torraca, Gabriel Gonçalves, Rogério Saturnino, Carlos Vilas Boas, Carlos Junior and Vitor Duarte. Thanks guys for always being there for me.

RESUMO

Cabral, Filipe Goulart. **The role of extreme points for convex hull operations**. 2018. 160 f. Dissertação (Mestrado em Matemática) - PGMat, Instituto de Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2018.

Esta dissertação trata de métodos de otimização convexa para problemas de otimização estocástica não convexos, e foi motivada pelos recentes desenvolvimentos no uso de variáveis binárias para problemas multi-estágio. Buscamos, aqui, apresentar de forma unificada dois resultados que estão no coração de dois algoritmos muito usados para programação não-convexa: o clássico teorema de Balas sobre o fecho convexo de uniões de poliedros, e o mais recente teorema da "bênção das variáveis binárias" de Zou, Ahmed e Sun, garantindo dualidade forte para programação estocástica com variáveis de estado binárias. Esta unificação será vista da forma mais geométrica que pudemos dar, interpretando ambos resultados em termo de produtos cartesianos e projeções. Esta mesma intuição geométrica será usada no momento de descrever novos modelos que se encaixem no arcabouço desta teoria.

Keywords: Programação Inteira Mista Estocástica, Programação Inteira Mista, Programação Disjuntiva, Análise Convexa.

ABSTRACT

Cabral, Filipe Goulart. **The role of extreme points for convex hull operations**. 2018. 160 f. Dissertação (Mestrado em Matemática) - PGMat, Instituto de Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2018.

This dissertation deals with convex optimization methods for non-convex stochastic optimization problems, and was motivated by recent developments for using binary variables in the multi-stage setting. Our text aims to present in a unified way two results which lie in the core of two widely used algorithms for non-convex programming: the classical Balas's Theorem about the convex hull of union of polihedra, and the more recent "Blessing of Binary" theorem from Zou, Ahmed and Sun, proving strong duality for stochastic programming with purely binary state variables. We strived for the most geometrical formulation, interpreting both results by means of cartesian products and projections. This same geometrical intuition will be used for describing new models that are amenable to this theory.

Keywords: Stochastic Mixed Integer Programming, Mixed Integer Programming, Disjunctive Programming, Convex Analysis.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Decision under uncertainty represents the heart of decision theory. This modeling paradigm takes into account all possible future outcomes when making a decision and how decisions over time affect each other. One such branch of decision under uncertainty is the Stochastic Programming field which assumes that the probability distribution of the underlying uncertainty is known[1], see SHAPIRO; DENTCHEVA; RUSZCZYŃSKI [26] and BIRGE; LOUVEAUX [7]. In this setting, the decision at a given point in time depends on past decisions and data observations, and the uncertainty are typically on the constraints and objective function parameters of a given mathematical program. Thus, a solution to a stochastic programming problem is a *decision rule* (also called policy), since we must obtain a function that returns a decision for a particular time and state of the system.

Modeling a real-life problem is a delicate balance between choosing the most relevant issues, having a mathematical framework available to represent the desired phenomenon, and having an algorithm able to solve the proposed formulation in a reasonable time. The main motivation of this study is the modeling limitation imposed by the convexity condition required in standard algorithms that solve large-scale multi-stage stochastic optimization programs such as the L-Shaped and the SDDP methods, see PEREIRA; PINTO [20], SHAPIRO [24], SHAPIRO; DENTCHEVA; RUSZCZYŃSKI [26], BIRGE; LOUVEAUX [7] and RUSZCZYNSKI [23]. The convexity requirement prevents the precise representation of several operational constraints that are relevant for the Brazilian power system operational planning such as the minimum reservoir outflow, unit commitment, AC power flow

[1]In practice, an expert fits a model for the uncertainty using statistical techniques, and in this work we assume that the fitted model describes perfectly our data.

constraints, and low storage volume risk aversion. With the increasing penetration of wind and solar generation, an accurate assessment of the physical and financial impacts of each decision is becoming even more relevant.

Recently, ZOU; AHMED; SUN [28] proposed the SDDiP algorithm to solve an important class of non-convex problems: the class of multistage stochastic integer programming (MSIP) problems. In light of this progress, we decided to investigate the theoretical and practical limits of the mixed integer modeling tool. We have found the Disjunctive Programming area (JEROSLOW [17], BALAS [1], BALAS [2]), which uses binary variables to model systems of linear inequalities joined by logical operators such as conjunctions ("and"), negations ("complement") and disjunctions ("or"), which motivated the name of Disjunctive Programming. Such logical constraints induce unions of polyhedra and the representation of this set using linear constraints and binary variables is performed by using an important formula referred in this dissertation as *Balas's formula*, which we introduce in chapter 5. Balas's formula is able to describe a large class of union of polyhedral sets, and when it fails, there is no set of linear constraints involving binary variables that is able to describe that given set.

Balas's formula also enjoys other properties. Actually, it is possible to compute the convex hull of a union of polyhedra by using that formula and considering the binary variables as continuous variables in the interval $[0, 1]$. This simplicity and efficiency for computing convex hulls gave rise to the Lift and Project algorithm (BALAS; CERIA; CORNUÉJOLS [3]) which is one of the tools presented in most commercial solvers for 0-1 mixed integer programs, see GOMORY [15] and CORNUÉJOLS [13] for details.

After studying the SDDiP and Balas's formula, we noticed a common phenomenon: both techniques take the convex hull of non-convex sets, but do not introduce new boundary points at certain regions after such operation. We prove that those boundary points are the extreme points of a particular convex set, and we call that property the "Blessing of extreme points". In particular, this result extends the "Blessing of Binary" of ZOU; AHMED; SUN [28] which states that for a function f of a class of mixed integer value functions, the corresponding convex regularization $\check{f}$ equals f at some notable points. This is an important theorem for analyzing the convergence of the SDDiP algorithm.

The whole development of this dissertation was motivated by geometric intuitions on convex sets. This is an approach presented in some convex programming books such as ROCKAFELLAR [22] and BERTSEKAS [5], which we frequently refer to in this work. An example of this geometric view is the study of functions by means of the epigraph, i.e., the set of points lying on or above its graph:

$$\text{epi}(f) = \{(x, w) \in \mathbb{R}^{n+1} \mid f(x) \leq w\}.$$

It is possible to show that a function f is convex if, and only if, its epigraph epi(f) is a convex set. This equivalence remains true even if f can assume the values $-\infty$ or $+\infty$. Another example is the interpretation of operations on functions in terms of operations on sets. The pointwise maxima of functions can be described by the intersection of epigraphs and partial minimization of a multivariate function can be represented by the projection of the epigraph. Moreover, this approach is also suitable for the analysis of some nonconvex problems such as mixed integer linear programming (MILP) problems, since the feasible sets of MILP problems are unions of polyhedra and the epigraph of optimal value functions of MILP problems are also unions of polyhedra.

We divided this dissertation into two parts: the necessary background of convex analysis and stochastic programming, which comprises chapters 2, 3 and 4, and the application of that theory to Disjunctive Programming, the SDDiP algorithm and the Blessing of Extreme Points, which are presented in chapters 5, 6, and 7, respectively. Below we describe in more details the content of each chapter.

- Chapter 2 is intended to prove the convex closure formula for the union of closed convex sets. We note that the proof of this result found in CERIA; SOARES [12] is wrong, which motivated us to write chapter 2 in greater detail. This formula is important for chapters 5 and 7. In a first reading, it is possible skip the proof of the main result of this chapter and just read the definitions presented in section 2.1.

- Chapter 3 analyzes the properties of optimal functions of mathematical programming problems. We use extended real-valued functions to unify the analysis of optimization problems that may be infeasible or infinite. Another advantage of this approach is the easiness of interpretation of every result in terms of geometrical operations in the epigraph of the analyzed function.

  Sections 3.1, 3.2 and 3.3 present some results that clarify important properties of the cost-to-go functions associated to the dynamic programming formulation of a multi-stage stochastic programming problem.

  Section 3.4 establishes a connection between the Lagrangian Relaxation of an optimization problem and the biconjugate operator for extended real valued functions. These are equivalent techniques to obtain convex approximations from non-convex optimal value functions. An important result of this section that will be used in Chapter 6 is the Lagrangian Relaxation formula for a mixed integer linear programming problem.

  Section 3.5 presents the subgradient notion, which extends the concept of derivative for non-differentiable convex functions. We demonstrate the chain rule for some operations that preserve convexity and the optimality condition in terms of subgradients. The main application of the notion of subgradient is the calculation of linear approximations, called cuts, used in both SDDP and SDDiP algorithms to estimate the cost-to-go function iteratively. We also prove that the optimal dual solutions of a given convex program are the subgradients of the corresponding primal optimal value function.

- Chapter 4 presents a brief introduction to stochastic optimization and the SDDP algorithm.

  Section 4.1 introduces the stochastic optimization formalism and the dynamic programming formulation for stagewise independent (SWI) processes.

  Section 4.2 introduces the notion of a SWI scenario tree to approximate the distribution of the underlying random process. In this case, the expected value in the future cost-to-go function definition becomes a weighted sum whose weights are the probability of the corresponding outcome. We comment on properties of the cost-to-go and future cost-to-go functions based on the theory developed in chapter 3.

  Section 4.3 presents the SDDP algorithm of PEREIRA; PINTO [20], which is a widely used algorithm for solving large-scale multi-stage stochastic programming problems. The description of the SDDP is crucial for understanding of ideas of SDDiP algorithms in chapter 6.

  Section 4.4 describes a long-term power system operational planning model used as a running examples through this dissertation.

- Chapter 5 deals with the theory of Disjunctive Constraints and their applications.

  Section 5.1 introduces the concept of Disjunctive Constraints and Balas's formula that allows the representation of certain union of polyhedra by linear constraints and mixed integer binary variables. We present an elementary proof of Jeroslow's theorem (JEROSLOW [18]) which states that if Balas's formula does not represent a given union of polyhedra then no set of linear constraints on mixed binary variables could represent it. Also in this section, we prove that the convex hull of the union of polyhedra can be obtained from Balas formula if we consider the binary variables as continuous in the interval $[0, 1]$. In Chapter 7 we investigate a common property between Balas formula and the SDDiP algorithm.

Section 5.2 presents nonconvex constraints that aim to describe operational rules more precisely and illustrates the use of Balas's formula. We introduce

(i) the risk aversion proposals of low stored volumes called CAR-B and SAR-B, which impose a minimum amount of thermal generation if the stored volume is outside a safe operating region;

(ii) the suppression of preventive deficit constraint, which makes infeasible any amount of deficit if the stored volume is not sufficiently low;

(iii) and the minimum outflow constraints which decreases the outflow requirement for low stored volumes.

- Chapter 6 contains our description of the SDDiP algorithm.

Section 6.1 presents some cut definitions for a non-convex function f. The Lagrangian cut is the most important among them, since it plays a central role for the description of the SDDiP algorithm. Given a function f, the Lagrangian cut is obtained from the subgradient of the convex regularization f^{**}, where f^{**} is the greatest closed convex function less than or equal to f;

Section 6.2 presents the Blessing of Binary theorem which states that the Lagrangian cuts are "tight" in the binary coordinates $\hat{\lambda} \in \{0, 1\}^p$ for particular class of optimal value functions ϕ of Mixed Integer Linear Programs (MILP), or equivalently, $\phi^{**}(\hat{\lambda}) = \phi(\hat{\lambda})$ for all $\hat{\lambda} \in \{0, 1\}^p$. This is the main theorem that explains how SDDiP estimates the cost-to-go functions of a multistage MILP stochastic program. Indeed, the SDDiP algorithm has the same forward and backward structure as the SDDP. However, it discretizes the state space, so it *only* computes feasible solutions in the forward step and Lagrangian cuts in the backwards step at the discrete states.

- Chapter 7 presents a connection between the SDDiP algorithm and Balas's formula for taking the convex hull of a finite union of polyhedra.

  Section 7.1 elaborates on Balas's Theorem for computing the convex hull by means of continuous relaxation of a binary variable. In some sense, Balas's formula can be seen as the union of Cartesian products $\bigcup_{i=1}^{p} P_i \times \{e_i\}$, where $e_1, \ldots, e_p$ are the vertices of a $p-1$ simplex. After taking the convex hull of $\bigcup_{i=1}^{p} P_i \times \{e_i\}$ the affine space $\mathbb{R}^n \times \{e_i\}$ still just have $P_i \times \{e_i\}$, that is,

$$\operatorname{conv}\left(\bigcup_{i=1}^{p} P_i \times \{e_i\}\right) \cap (\mathbb{R}^n \times \{e_i\}) = \bigcup_{i=1}^{p} P_i \times \{e_i\}.$$

  This property is not exclusive for Balas's formula, it also holds for unions $\bigcup_{i=1}^{p} P_i \times \{r_i\}$, where $r_1, \ldots, r_p$ are extreme points of an arbitrary convex set. This is the discrete version of the "Blessing of extreme points".

  Section 7.2 revisits the Blessing of Binary theorem of ZOU; AHMED; SUN [28] and presents a geometric perspective for this result regarding a larger class of functions: the piecewise proper closed convex functions. Denote by f a function of this class and by f^{**} the corresponding convex regularization. We prove that if a pair $(x, f^{**}(x))$ is an extreme point of $\operatorname{epi}(f^{**})$, then $f^{**}(x) = f(x)$. Since, in general, we do not have an explicit expression for the convex regularization of f, it can be very difficult to check the condition of such theorem. However, we also prove that if x is an extreme point of $\operatorname{dom}(f)$, then $(x, f^{**}(x))$ is also an extreme point of $\operatorname{epi}(f^{**})$. This result comprises the Blessing of Binary theorem and extends it to a larger class of non-convex functions.

2 CONVEX HULL OF UNION OF CLOSED CONVEX SETS

In this chapter, we prove some important facts concerning the convex hull of union of closed convex sets. Those are the key arguments to understand our main result: the Blessing of Extreme Points.

It is instructive to note that the convex hull of union of closed convex sets may not be closed, see figures 2.7a and 2.7b (page 28). However, for the particular class of closed convex sets, we have an enlightening identity. Under mild regularity conditions detailed in Theorem 2.10 (page 27), we have that

$$\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{m} D_i\right) = \operatorname{conv}\left(\bigcup_{i=1}^{m} D_i\right) + \sum_{i=1}^{m} \operatorname{recc}(D_i), \tag{2.1}$$

where $D_1, \ldots, D_m$ are nonempty closed convex sets of $\mathbb{R}^n$, and $\operatorname{recc}(D_i)$ is the recession cone of D_i, which contains all directions $d_i \in \mathbb{R}^n$ such that starting at any $x_i \in D_i$ and going indefinitely along d_i, we never leave D_i. Thus, if the set $\operatorname{conv}\left(\cup_{i=1}^{m} D_i\right)$ is not closed, we simply add the recession directions to make it closed, see figures 2.7b and 2.7c (page 28), for an example. Moreover, if each D_i is a polyhedral set, then formula (2.1) holds without any further regularity condition.

The purpose of the following sections is to build the necessary theory to establish identity (2.1). We have chosen to carefully prove (2.1), since we have found a critical mistake in the proof of CERIA; SOARES [12] (page 601). We provide an original proof for (2.1) under the suitable regularity conditions. On a first reading, it is possible to go through section 2.1 for basic results in convex analysis, and use equation (2.1) when necessary.

We organize the exposition as:

1. In section 2.1, we present some basic results on convex analysis;
2. In section 2.2, we prove some properties of sum and closure of lifted cones $K = \text{cone}(\{1\} \times D)$;
3. In section 2.3, we show the conditions for commuting a given linear map A with the closure operation, that is, when $\text{cl}(AD) = A\,\text{cl}(D)$;
4. In section 2.4, we conclude that the equality $\text{cl}\left(\sum_{i=1}^{m} K_i\right) = \sum_{i=1}^{m} \text{cl}(K_i)$ implies the desired equation (2.1), where $K_i = \text{cone}(\{1\} \times D_i)$.

2.1 Basic results on convex analysis

In this section, we introduce some basic definitions and results of convex analysis that will be important throughout this work.

Let D be a set of $\mathbb{R}^n$. We say that D is *convex* if for any two points of D, the line segment that joins them is also within D, see figure 2.1. The set $P \subset \mathbb{R}^n$ is a *polyhedron* if it is the intersection of a finite number of hyperspaces $a_j^\top x \leq b_j$, that is,

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\},$$

for some $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. In particular, P is a convex set, since any hyperspace is a convex set and the intersection of convex sets is convex. We say that $C \subset \mathbb{R}^n$ is a *cone* if for each vector $x \in C$ and scalar $\alpha \geq 0$, the product αx belongs to C.

A vector d is a *direction of recession* of D if $x + td$ belongs to D, for all $x \in D$

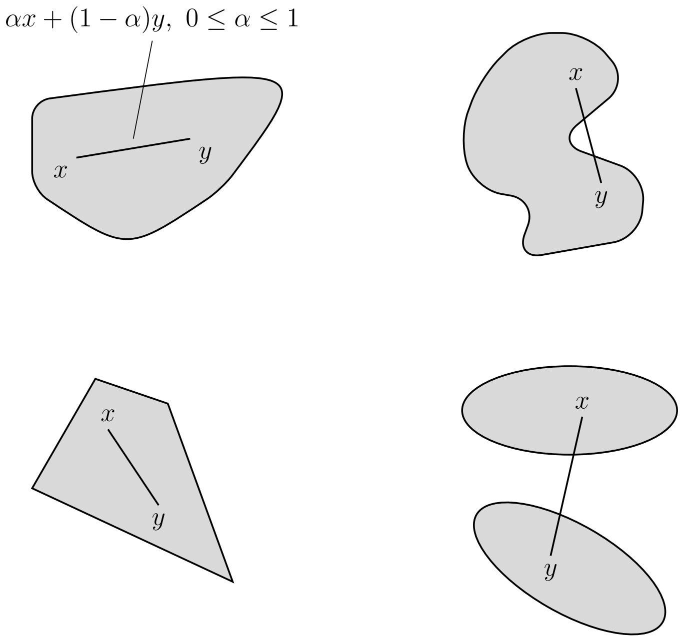


Figure 2.1: Convex (left side) and non-convex (right side) sets.

and $t \geq 0$. In other words, d is a direction of recession of D if starting at any x in D and going indefinitely along d, we never leave the set D. See figure 2.2 for an illustration. The set of all recession directions of D is denoted by $\mathrm{recc}(D)$ and it is a

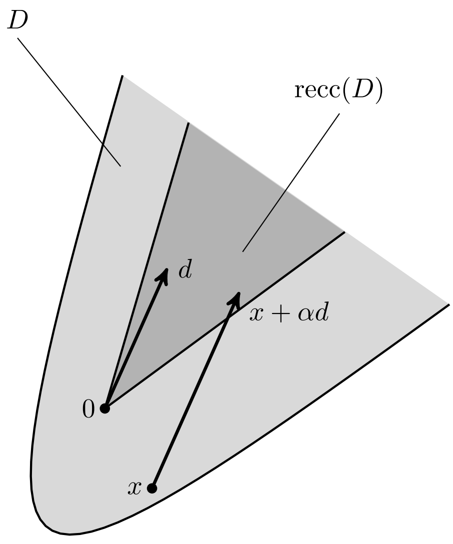


Figure 2.2: Recession cone of a convex set.

convex cone containing the origin. We note that the recession cone of a convex cone is the cone itself. The following result regarding the directions of recession states that if one can go indefinitely along d starting at one given point, we can go along d starting at any point, so d is a direction of recession. For a proof, see BERTSEKAS [5, page 43].

Theorem 2.1 (Recession cone theorem)**.** *Let D be a nonempty closed convex set.*

a) The recession cone $\operatorname{recc}(D)$ *is a nonempty closed convex cone, and is equal to* $\{0\}$ *if and only if D is compact;*

b) The vector d belongs to $\operatorname{recc}(D)$ *if and only if there is a vector $x \in D$ such that $x + td$ belongs to D, for every $t \geq 0$.*

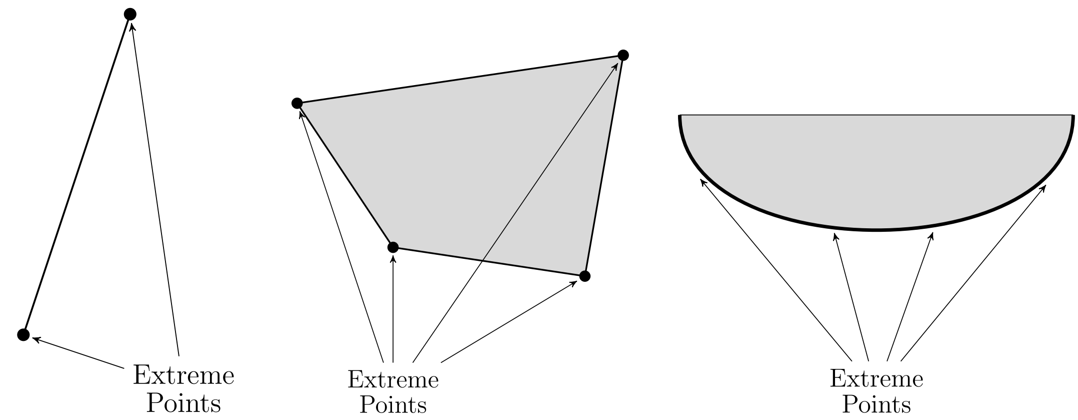
Given a nonempty convex set D, we say that $x \in D$ is an *extreme point* of D if it does not lie strictly between the endpoints of any line segment contained in D. In other words, the vector x is an extreme point of D if given any representation of x as a convex combination of $y, z \in D$, we have that x is equal to y or z. For an illustration, see figure 2.3a, 2.3b and 2.3c.

The *convex hull* of a set X, denoted by $\operatorname{conv}(X)$, is the set of all possible convex combinations of points of X, that is,

$$\operatorname{conv}(X) = \left\{ \sum_{i=1}^{k} \lambda_i x_i \in \mathbb{R}^n \;\middle|\; \begin{array}{cc} x_i \in X, & \sum_{i=1}^{k} \lambda_i = 1, \quad \lambda_i \geq 0, \\ 1 \leq i \leq k, & k \in \mathbb{N}. \end{array} \right\}.$$

One can prove that the convex hull of X is the smallest convex set containing X. The *conical hull* of a set X, denoted by $\operatorname{cone}(X)$, is the set of all possible positive combinations of points of X, that is,

$$\operatorname{cone}(X) = \left\{ \sum_{i=1}^{k} \alpha_i x_i \in \mathbb{R}^n \;\middle|\; \begin{array}{cc} x_i \in X, & \alpha_i \geq 0, \\ 1 \leq i \leq k, & k \in \mathbb{N}. \end{array} \right\}.$$



(a) Extremes of a line segment. (b) Extremes of a polyhedra. (c) Extremes of a convex set.

Figure 2.3: Extreme points of convex sets.

It is also true that cone(X) is the smallest convex cone containing X.

An important theorem regarding polyhedral sets is the Minkowski-Weyl representation theorem [5, page 106]: every polyhedron P is the Minkowski sum of the convex hull of a finite number of points and the conic hull of another finite set of points, as exemplified in figure 2.4.

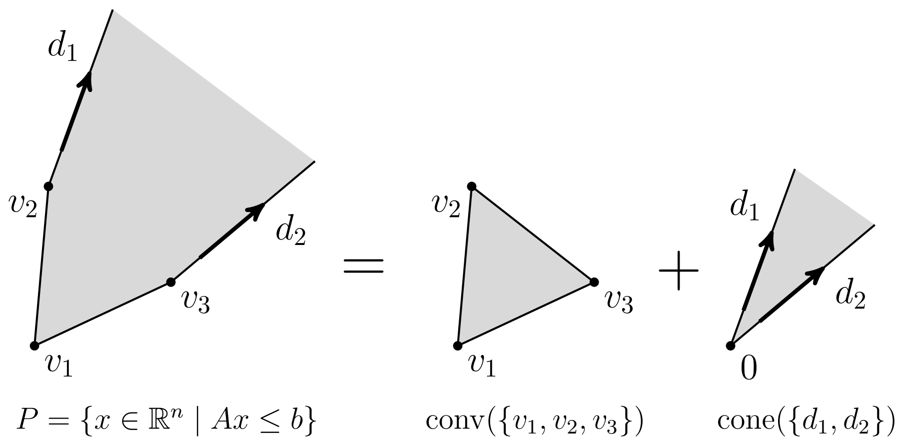


Figure 2.4: Minkowski-Weyl Theorem.

Theorem 2.2 (Minkowski-Weyl representation)**.** *The set P is polyhedral if and only* *if there is a finite number of vectors* $\{v_1, \ldots, v_k\} \subset \mathbb{R}^n$ *and* $\{w_1, \ldots, w_l\} \subset \mathbb{R}^n$ *such that*

$$P = \text{conv}(\{v_1, \ldots, v_k\}) + \text{cone}(\{w_1, \ldots, w_l\}).$$

In particular, the recession cone of P *is equal to* $\text{cone}(\{w_1, \ldots, w_l\})$*.*

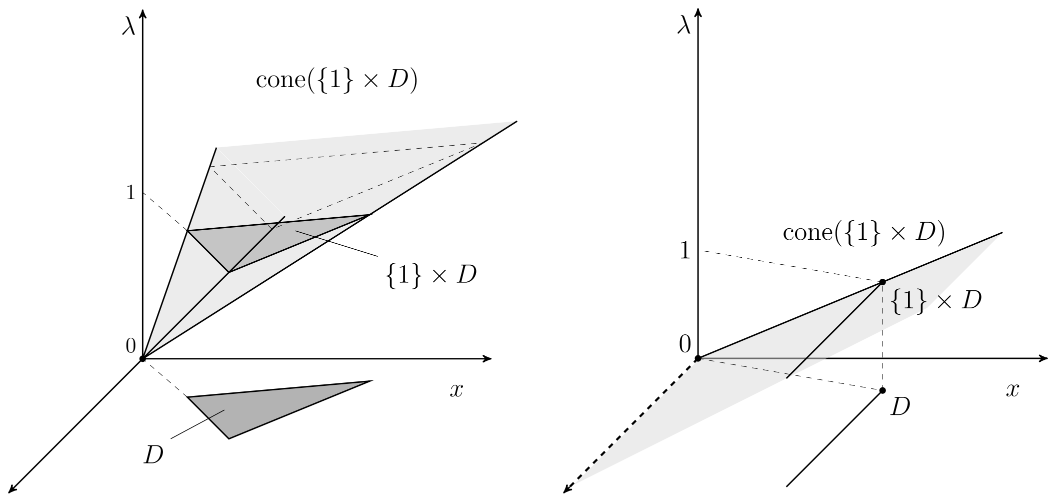
The Minkowski-Weyl representation is an important theorem since several complicated results for the general convex case can be easily demonstrated in the polyhedral case thanks to this theorem without any further regularity condition.

2.2 Conical lifting

We define the *conical lift* of a set $D \subset \mathbb{R}^n$ as the conic hull K of the particular set $\{1\} \times D$, as depicted in figures 2.5a and 2.5b. It is instructive to note that K is *always* convex, independently of the convexity of D, since K is the set of all positive combinations of elements of $\{1\} \times D$. The set K is equal to the set $\{(\lambda, \lambda x) \in \mathbb{R}^{n+1} \mid x \in D, \lambda \geq 0\}$ if, and only if, the set D is convex.

The advantage of transforming a convex set into this special convex cone is that the convex hull of the union of convex sets is closely related to the sum of the corresponding lifted cones, as we will see in Corollary 2.5, page 18.

Another important result for us is the characterization of the closure of the lift cone. For any unbounded closed convex set D, the $\text{cone}(\{1\} \times D)$ is not closed, and the missing boundary set is $\{0\} \times \text{recc}(D)$, see figure 2.5b. Let $D_1, \ldots, D_m$ be nonempty closed convex sets, and let $K_i = \text{cone}(\{1\} \times D_i)$ be the corresponding conical lifts. We shall see that the closure of the convex hull of $\cup_{i=1}^m D_i$ can be obtained from the closure of the sum of lifted cones $\sum_{i=1}^m K_i$. Thus, formula (2.1)



(a) Conical lift of a compact set (b) Conical lift of an unbounded set

Figure 2.5: Conical lifting.

is proved by noting that, under some regularity conditions, the closure commutes with the sum:

$$\mathrm{cl}\left(\sum_{i=1}^{m} K_i\right) = \sum_{i=1}^{m} \mathrm{cl}(K_i).$$

The following lemma characterizes the recession directions of a given closed convex set D in terms of sequences. This is useful to prove that the closure of the corresponding lifted cone K is equal to $K \cup (\{0\} \times \mathrm{recc}(D))$.

Lemma 2.3. *(Diverging sequence on a closed convex set) Let D be a nonempty closed convex set, and let $\{x^k\}_{k\in\mathbb{N}}$ be a diverging sequence in D. Then, every accumulation point of $\{x^k/\|x^k\|\}_{k\in\mathbb{N}}$ belongs to the recession cone* recc(D)*:*

$$\lim_{j\to+\infty} \frac{x^{k_j}}{\|x^{k_j}\|} = d \in \mathrm{recc}(D).$$

Proof. We can suppose that $\lim\limits_{k\to\infty} x^k/\|x^k\| = d$. Now, let t be a positive scalar and note that $\lim_{k\to\infty}(x^k - x^1)/\|x^k - x^1\| = d$ since $\|x^k\|$ is unbounded:

$$\lim_{j\to\infty} \frac{(x^{k_j} - x^1)}{\|x^{k_j} - x^1\|} = \lim_{j\to\infty} \left(\frac{x^{k_j}}{\|x^{k_j}\|} \frac{\|x^{k_j}\|}{\|x^{k_j} - x^1\|} - \frac{x^1}{\|x^{k_j} - x^1\|} \right) = \lim_{j\to\infty} \frac{x^{k_j}}{\|x^{k_j}\|} = d.$$

Denote by d^k the vector given by $(x^k - x^1)/\|x^k - x^1\|$, and by t^k the scalar $\|x^k - x^1\|$. Since $\|x_k\|$ diverges, there is $N \in \mathbb{N}$ such that t_k is greater than t for every k greater than N. Note that $x^1 + t^k d^k$ belongs to D for every k,

$$x^k = x^1 + t^k d^k \in D, \quad \forall k \in \mathbb{N},$$

and $x^1 + td^k$ also belongs to D, for all k greater than N, since it is a convex combinations of x^1 and $x^1 + t^k d^k$. So, $x^1 + td^k$ converges to $x^1 + td$, which belongs to D since D is a closed set. Therefore, $x^1 + td$ belongs to D for every positive scalar t, that is, d is a direction of recession from D. □

The following theorem formalizes the intuition about taking the closure of a cone $K = \operatorname{cone}(\{1\} \times D)$, where D is a convex set.

Theorem 2.4 (Closure of lift cone)**.** *Let D be a nonempty convex set, and let K be the conic hull of the Cartesian product $\{1\} \times D = \{(1, x) \in \mathbb{R}^{n+1} \mid x \in D\}$, that is, $K = \operatorname{cone}(\{1\} \times D)$. Denote by $\overline{D}$ the closure of D. Then, the closure of K is given by*

$$\operatorname{cl} K = \operatorname{cone}(\{1\} \times \overline{D}) \cup (\{0\} \times \operatorname{recc}(\overline{D})).$$

Proof. Let C_0, C_1, K_0 and K_+ be the following sets:

$$C_0 = \{0\} \times \operatorname{recc}(\overline{D}) \qquad K_0 = \operatorname{cl} K \cap \big(\{0\} \times \mathbb{R}^n\big)$$
$$C_1 = \operatorname{cone}(\{1\} \times \overline{D}) \qquad K_+ = \operatorname{cl} K \cap \big((0, \infty) \times \mathbb{R}^n)\big)$$

We need to prove that C_0 equals K_0 and $C_1 \backslash \{(0, 0)\}$ equals K_+. This is sufficient to prove this theorem, since $\operatorname{cl} K = K_+ \cup K_0$.

Let's start by proving the equality between K_0 and C_0. Let $y \in K_0$. Then, y is equal to $(0, d)$ for some $d \in \mathbb{R}^n$ and there is a sequence $\{y_k\}_{k\in\mathbb{N}}$ from K that converges to y. For each $y_k \in K$, there is $\lambda_k \geq 0$ and $x_k \in D$ such that $y_k = \lambda_k(1, x_k)$. Thus, we have that

$$\lim_{k\to\infty} \lambda_k = 0, \quad \lim_{k\to\infty} \lambda_k x_k = d.$$

If d is 0, then y belongs to C_0. If d is different from 0, then $\|x_k\|$ goes to infinity:

$$\lim_{k\to\infty} \|x_k\| = \lim_{k\to\infty} \frac{\|\lambda_k x_k\|}{\lambda_k} = \frac{\|d\|}{0^+} = +\infty.$$

Denote by d_k the vector $x_k/\|x_k\|$. From Lemma 2.3, every convergent subsequence of $\{d_k\}_{k\in\mathbb{N}}$ converges to a vector u that belongs to the recession cone $\mathrm{recc}(\overline{D})$. Taking a subsequence if necessary, we have the following limit:

$$d = \lim_{k\to\infty} \lambda_k x_k = \lim_{k\to\infty} \|\lambda_k x_k\| \frac{x_k}{\|x_k\|} = \|d\| u.$$

Thus, d is also a recession direction of $\overline{D}$, and y belongs to C_0. Therefore, $K_0 \subseteq C_0$.

Let's prove the opposite inclusion. Let $y \in C_0$. If y is equal to $(0, 0)$, then y belongs to K_0, trivially. Suppose that y is equal to $(0, \overline{d})$, where $\overline{d}$ is some non-zero vector of $\mathrm{recc}(\overline{D})$. Let $\overline{x}_0 \in \overline{D}$ and note that $\overline{x}_k = x_0 + k\overline{d}$ belongs to $\overline{D}$, for every $k \in \mathbb{N}$. Let $x_k \in D$ be such that $\|x_k - \overline{x}_k\| \leq 1$, and let $y_k = \frac{1}{k}(1, x_k)$. Note that y_k belongs to K, for every $k \in \mathbb{N}$, and

$$\lim_{k\to\infty} y_k = \lim_{k\to\infty} \left[\frac{1}{k}(1, x_k - \overline{x}_k) + \frac{1}{k}(1, \overline{x}_k)\right] = (0, 0) + (0, \overline{d}) = y.$$

So, y belongs to K_0. Therefore, $K_0 = C_0$.

Now, let's prove the equality $K_+ = C_1 \backslash \{(0, 0)\}$. Let $y \in K_+$. Then, y is equal to (λ, x) for some $\lambda > 0$ and $x \in \mathbb{R}^n$, and there is a sequence $\{y_k\}_{k\in\mathbb{N}}$ from K such that $\lim\limits_{k\to\infty} y_k = (\lambda, x)$. Thus, $y_k = \lambda_k(1, x_k)$ for some $\lambda_k \geq 0$ and $x_k \in D$. Note that

$$\lim_{k\to\infty} x_k = \lim_{k\to\infty} \frac{1}{\lambda_k}(\lambda_k x_k) = \frac{1}{\lambda} x \in \overline{D}.$$

So, $y = \lambda\left(1, \frac{x}{\lambda}\right)$ belongs to $C_1\backslash\{(0,0)\}$. Therefore, $K_+ \subseteq C_1\backslash\{(0,0)\}$.

Let's prove the opposite inclusion. Let $y \in C_0\backslash\{(0,0)\}$. Then, $y = \lambda(1, \overline{x})$ for some $\lambda > 0$ and $\overline{x} \in \overline{D}$. Let $\{x_k\}_{k\in\mathbb{N}}$ be a sequence from D that converges to x, and denote by y_k the vector $\lambda(1, x_k)$. Therefore, y_k belongs to K for every k, and we have that

$$\lim_{k\to\infty} y_k = (\lambda, \lambda \lim_{k\to\infty} x_k) = (\lambda, \lambda x) = y.$$

So, y belongs to K_+. Therefore, $K_+ = C_1\backslash\{(0,0)\}$. □

One advantage of the conical lift is to transform the convex hull of a union of convex sets D_i into a sum of convex cones $K_1 + \cdots K_m$, where $K_i = \mathrm{cone}(\{1\} \times D_i)$. Moreover, by adding the closure of the cones $\mathrm{cl}\, K_1 + \cdots + \mathrm{cl}\, K_m$ we obtain an expression that contains the right-hand side of equation (2.1): $\mathrm{conv}(\bigcup_{i=1}^m D_i) + \sum_{i=1}^m \mathrm{recc}(D_i)$.

Corollary 2.5 (Convex hull by conical lift)**.** *Let $D_1, \ldots, D_m$ be nonempty closed convex sets, let D be the union $\cup_{i=1}^m D_i$, let R be the sum of recession cones $\sum_{i=1}^m \mathrm{recc}(D_i)$, and let K_i be the conic hull of $\{1\} \times D_i$, for all $i = 1, \ldots, m$. Then,*

$$\begin{aligned} K_1 + \cdots + K_m &= \mathrm{cone}\big(\{1\} \times \mathrm{conv}(D)\big), \\ \mathrm{cl}\, K_1 + \cdots + \mathrm{cl}\, K_m &= \mathrm{cone}\big(\{1\} \times [\mathrm{conv}(D) + R]\big) \cup \big(\{0\} \times R\big). \end{aligned}$$

Proof. Let K be the sum of cones $K_1, \ldots, K_m$, that is, $K = \sum_{i=1}^m K_i$. We shall prove that K is equal to $\mathrm{cone}\big(\{1\} \times \mathrm{conv}(D)\big)$. Indeed, let y be a vector of K. Then, there is a scalar $\lambda \geq 0$ and a vector $u \in \mathbb{R}^n$ such that $y = (\lambda, u)$ and

$$(\lambda, u) = \sum_{i=1}^m \lambda_i (1, x_i),$$

for some $\lambda_i \geq 0$, and $x_i \in D_i$, for $i = 1, \ldots, m$. If λ is equal to 0, then each λ_i is equal to 0. In particular, y is equal to 0 and it belongs to $\mathrm{cone}\big(\{1\} \times \mathrm{conv}(D)\big)$

trivially. Suppose that λ is positive. Then,

$$y = \lambda \sum_{i=1}^{m} \frac{\lambda_i}{\lambda}(1, x_i) = \lambda \left(1, \sum_{i=1}^{m} \gamma_i x_i\right),$$

where $\gamma_i := \lambda_i/\lambda$. Therefore, y belongs to $\operatorname{cone}\big(\{1\} \times \operatorname{conv}(D)\big)$, since $\sum \gamma_i = \sum \lambda_i/\lambda = \lambda/\lambda = 1$. On the other hand, let y be a vector of $\operatorname{cone}\big(\{1\} \times \operatorname{conv}(D)\big)$. Then, there are scalars $\lambda \geq 0$, $\gamma_j \in [0,1]$, and vectors $x_j \in D_j$ such that $\sum_{i=1}^{m} \gamma_i = 1$ and

$$y = \lambda \left(1, \sum_{i=1}^{m} \gamma_i x_i\right) = \sum_{i=1}^{m} \lambda \gamma_i (1, x_i),$$

for all $j = 1, \ldots, m$. Therefore, y belongs to K, and we have that K is equal to $\operatorname{cone}\big(\{1\} \times \operatorname{conv}(D)\big)$.

We now prove the second equality. Let $\widetilde{K}$ be the sum of cones $\operatorname{cl}(K_1), \ldots,$ $\operatorname{cl}(K_m)$, i.e., $\widetilde{K} = \sum_{i=1}^{m} \operatorname{cl} K_i$. Then, the second equality can be stated as

$$\widetilde{K} = \operatorname{cone}\big(\{1\} \times [\operatorname{conv}(D) + R]\big) \cup \big(\{0\} \times R\big).$$

Now, let y be a vector of $\widetilde{K}$. Since y belongs to $\widetilde{K}$, there is a scalar $\lambda \in \mathbb{R}$ and a vector $u \in \mathbb{R}^n$ such that $y = (\lambda, u)$ and

$$(\lambda, u) = \sum_{i=1}^{m} (\lambda_i, u_i),$$

for some $(\lambda_i, u_i) \in \operatorname{cl}(K_i)$, for $i = 1, \ldots, m$. From Theorem 2.4 and since D_i are closed, the closure of K_i is equal to $K_i \cup \{0\} \times \operatorname{recc}(D_i)$. Let I^+ be the set of indexes i such that λ_i is greater than zero, and let I^0 be the set of indexes i such that λ_i is equal to zero. For each $i \in I^+$, there is $x_i \in D_i$ such that u_i is equal to $\lambda_i x_i$, and for each $i \in I^0$, there is $d_i \in \operatorname{recc}(D_i)$ such that u_i is equal to d_i. Then,

$$(\lambda, u) = \sum_{i \in I^+} \lambda_i (1, x_i) + \sum_{i \in I^0} (0, d_i).$$

If λ is equal to zero, then each λ_i is equal to zero. Therefore, the vector y belongs to $\{0\} \times R$. If λ is positive, denote by γ_i the ratio λ_i/λ. Then, we have that

$$\begin{aligned}(\lambda, u) &= \lambda \sum_{i \in I^+} \gamma_i (1, x_i) + \lambda \sum_{i \in I^0} \left(0, \frac{1}{\lambda} d_i\right) \\ &= \lambda \left(1, \sum_{i \in I^+} \gamma_i x_i + \sum_{i \in I^0} \frac{1}{\lambda} d_i\right),\end{aligned}$$

and so y belongs to cone $\big(\{1\} \times [\mathrm{conv}(D) + R]\big)$. Lets prove the opposite inclusion. First, note that the inclusion $(\{0\} \times R) \subset (\mathrm{cl}(K_1) + \cdots + \mathrm{cl}(K_m))$ follows from the closure formula for $\mathrm{cl}(K_i)$, that is, $\mathrm{cl}(K_i) = K_i \times (\{0\} \times \mathrm{recc}(D_i))$. Let y be a vector of the set cone $\big(\{1\} \times [\mathrm{conv}(D) + R]\big)$. Then, y has the form $(\lambda, u) \in \mathbb{R}^{n+1}$ such that

$$(\lambda, u) = \lambda \left(1, \sum_{i=1}^{m} \gamma_i x_i + \sum_{i=1}^{m} d_i\right),$$

for some $x_i \in D_i$, $d_i \in \mathrm{recc}(D_i)$, and $\gamma_i \in [0, 1]$ with $\sum_{j=1}^{m} \gamma_j = 1$, for $i = 1, \dots, m$. Let I^+ and I^0 be the set of indexes i for which γ_i is equal to zero and positive, respectively. Then, we can split the sum $\sum_{i=1}^{m} d_i$ into $\sum_{i \in I^+} d_i$ and $\sum_{i \in I^0} d_i$, and we have that

$$\begin{aligned}(\lambda, u) &= \lambda \left(1, \sum_{i \in I^+} \gamma_i \left(x_i + \gamma_i^{-1} d_i\right) + \sum_{i \in I^0} d_i\right) \\ &= \sum_{i \in I^+} \lambda \gamma_i (1, x_i + \gamma_i^{-1} d_i) + \sum_{i \in I^0} (0, \lambda d_i).\end{aligned}$$

Note that $x_i + \gamma_i^{-1} d_i \in D_i$, for all $i \in I^+$, and $\lambda d_i \in \mathrm{recc}(D_i)$, for all $i \in I^0$. Thus, it follows that

$$\begin{aligned}\lambda \gamma_i (1, x_i + \gamma_i^{-1} d_i) &\in \mathrm{cl}(K_i), \quad \forall i \in I^+, \\ (0, \lambda d_i) &\in \mathrm{cl}(K_i), \quad \forall i \in I^0.\end{aligned}$$

Therefore, y belongs to $\widetilde{K}$. $\square$

2.3 Closedness under linear transformation

A common mistake in convex analysis regarding a closed convex set D is to assume that the image of D by a linear map A remains closed and convex. It is straightforward from the definition to prove that $A(D)$ is convex, but Figure 2.6 provides a counterexample for the closedness property. In this section, we will see

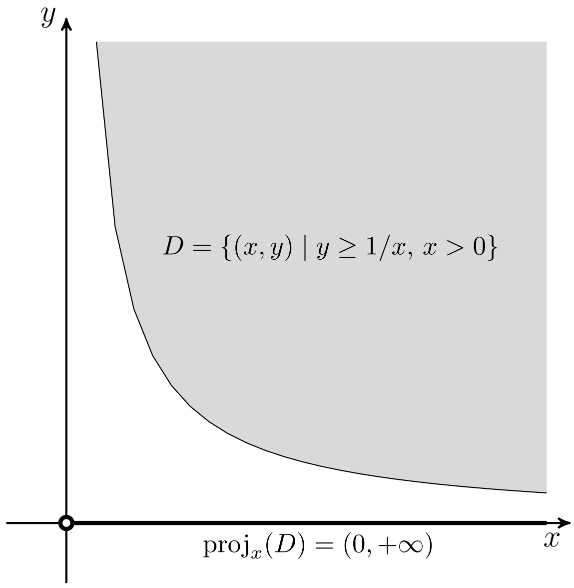


Figure 2.6: Non-closed image of closed convex set under linear map.

sufficient conditions on the linear map A and the closed convex sets D which assure that the image of D by A is a closed set. Theorem 2.6 below is the main result of this section.

Theorem 2.6 (Closure of image under linear transformation). *Let D be a nonempty convex set, and let A be a linear map from $\mathbb{R}^n$ to $\mathbb{R}^l$. Suppose that every vector d belonging to the intersection between the recession cone of* $\text{cl}(D)$ *and the kernel of A, i.e.,* $Ad = 0$*, also has the opposite direction $-d$ belonging to the recession cone* *of* $\mathrm{cl}(D)$*, i.e.,* $\mathrm{recc}(\mathrm{cl}(D)) \cap \ker(A) \subseteq -\mathrm{recc}(\mathrm{cl}(D))$*. Then,*

$$\mathrm{cl}(AD) = A(\mathrm{cl}(D)),$$
$$\mathrm{recc}(\mathrm{cl}(AD)) = A(\mathrm{recc}(\mathrm{cl}(D))).$$

In particular, if D *is closed then* AD *is also closed.*

Proof. Let y be a vector of $A(\mathrm{cl}(D))$. So, y is equal to Ax, for some $x \in \mathrm{cl}(D)$. Since x is an adherent point of D, there is a sequence $\{x_k\}_{k\in\mathbb{N}}$ from D that converges to x. Denote by y_k the vector Ax_k. Note that y_k belongs to AD, for all $k \in \mathbb{N}$, and $\{y_k\}_{k\in\mathbb{N}}$ converges to y:

$$\lim_{k\to\infty} y_k = \lim_{k\to\infty} Ax_k = Ax = y.$$

Therefore, y belongs to $\mathrm{cl}(AD)$, so we conclude that $A\,\mathrm{cl}(D) \subset \mathrm{cl}(AD)$.

Let y be a vector of $\mathrm{cl}(AD)$, and let $\{y_k\}_{k\in\mathbb{N}}$ be a sequence from AD that converges to y. Let x_k be a vector of $\mathrm{cl}(D)$ with minimum norm such that $Ax_k = y_k$, i.e., let x_k be the optimal solution of the following problem:

$$\begin{array}{rl} \min_x & \|x\| \\ \text{s.t.} & Ax = y_k, \\ & x \in \mathrm{cl}\, D. \end{array}$$

Note that x_k is unique, since the euclidean norm is a strictly convex function. Thus, there are two possibilities for the sequence $\{x_k\}_{k\in\mathbb{N}}$:

1. The sequence $\{x_k\}_{k\in\mathbb{N}}$ is bounded. Then, there is a subsequence $\{x_{k_j}\}_{j\in\mathbb{N}}$ that converges to some $x \in \mathrm{cl}(D)$. Then,

$$y = \lim_{j\to\infty} y_{k_j} = \lim_{j\to\infty} Ax_{k_j} = Ax,$$

so y belongs to $A(\mathrm{cl}(D))$.

2. The sequence $\{x_k\}_{k\in\mathbb{N}}$ has an unbounded subsequence. Without loss of generality, suppose that $\{x_k\}_{k\in\mathbb{N}}$ is unbounded, and let d be a cluster point of $\{x_k/\|x_k\|\}_{k\in\mathbb{N}}$. Taking a subsequence if necessary, assume that $\{x_k/\|x_k\|\}_{k\in\mathbb{N}}$ converges to d. From Lemma 2.3, d is a direction of recession of $\mathrm{cl}(D)$, and d also belongs to the kernel of A:
$$\|Ad\| = \lim_{k\to\infty} \frac{\|Ax_k\|}{\|x_k\|} = \lim_{k\to\infty} \frac{\|y_k\|}{\|x_k\|} = \frac{\|y\|}{+\infty} = 0.$$
So, by hypothesis, the opposite direction $-d$ is a direction of recession of $\mathrm{cl}(D)$. Denote by $\tilde{x}_k$ the vector $x_k - \|x_k\|d$. Note that $\tilde{x}_k \in \mathrm{cl}(D)$, and $A\tilde{x}_k = y_k$, for all $k \in \mathbb{N}$. In particular, the sequence $\{\tilde{x}_k/\|x_k\|\}_{k\in\mathbb{N}}$ converges to 0:
$$\lim_{k\to\infty} \frac{\|\tilde{x}_k\|}{\|x_k\|} = \lim_{k\to\infty} \frac{\|(x_k - \|x_k\|d)\|}{\|x_k\|} = 0.$$
So, for sufficiently large k, $\|\tilde{x}_k\| < x_k$, contradicting the hypothesis of x_k being a vector of minimum norm such that $x_k \in \mathrm{cl}(D)$ and $Ax_k = y_k$. Therefore, the sequence $\{x_k\}_{k\in\mathbb{N}}$ is bounded, and this proves that $\mathrm{cl}(AD) = A(\mathrm{cl}\, D)$.

We now prove the formula for the recession cone
$$\mathrm{recc}(\mathrm{cl}(AD)) = A(\mathrm{recc}(\mathrm{cl}(D))).$$

The idea for this proof is to take the conical lift K of D, extend the linear map $A \in \mathbb{R}^{n\times l}$ to a linear map $B = \begin{bmatrix} 1 & 0 \\ 0 & A \end{bmatrix} \in \mathbb{R}^{(n+1)\times(l+1)}$, and conclude that the closure operation and the linear map B commute, that is, $\mathrm{cl}(BK) = B\,\mathrm{cl}(K)$. Indeed, from Theorem 2.4 (page 16), we have the following expression for $B\,\mathrm{cl}(K)$:
$$\begin{aligned} B\,\mathrm{cl}(K) &= B\Big(\mathrm{cone}\left(\{1\} \times \mathrm{cl}(D)\right) \cup \left(\{0\} \times \mathrm{recc}(\mathrm{cl}(D))\right)\Big) \\ &= \mathrm{cone}\left(\{1\} \times A\,\mathrm{cl}(D)\right) \cup \left(\{0\} \times A\,\mathrm{recc}(\mathrm{cl}(D))\right). \end{aligned}$$

On the other hand, the set $\mathrm{cl}(BK)$ has the following form:
$$\begin{aligned} \mathrm{cl}(BK) &= \mathrm{cl}\Big(B\,\mathrm{cone}\left(\{1\} \times D\right)\Big) = \mathrm{cl}\Big(\mathrm{cone}\left(\{1\} \times AD\right)\Big) \\ &= \mathrm{cone}\left(\{1\} \times \mathrm{cl}(AD)\right) \cup \left(\{0\} \times \mathrm{recc}(\mathrm{cl}(AD))\right), \end{aligned}$$

where the last equality follows from Theorem 2.4, page 16.

If the closure operation commutes with the linear map B, i.e., if $\mathrm{cl}(BK) = B\,\mathrm{cl}(K)$, taking intersection of both sets with $\{0\} \times \mathbb{R}^n$ we obtain

$$\{0\} \times A\,\mathrm{recc}(\mathrm{cl}(D)) = \{0\} \times \mathrm{recc}(\mathrm{cl}(AD)),$$

which yields the required formula for the recession cone. So, it is enough to show that $\mathrm{cl}(BK) = B\,\mathrm{cl}(K)$, and for this purpose we use the first part of this theorem.

Note that the kernel of B is equal to $\{0\} \times \ker(A)$, and the recession cone of $\mathrm{cl}(K)$ is equal to itself, since the recession cone of a convex cone is the cone itself. So,

$$\begin{aligned} \ker(B) \cap \mathrm{cl}(K) &= \{0\} \times \big(\ker(A) \cap \mathrm{recc}(\mathrm{cl}(D))\big) \\ &\subset \{0\} \times \big(-\mathrm{recc}(\mathrm{cl}(D))\big) \\ &\subset -\mathrm{cl}(K), \end{aligned}$$

where the second relation follows from the hypothesis, and the first and third follow from the expression of $\mathrm{cl}(K)$ (Theorem 2.4). Using the first part of this theorem, we conclude that $\mathrm{cl}(BK) = B\,\mathrm{cl}(K)$. □

There are even more general conditions, such as the notion of convex retractive sets (see BERTSEKAS [5], page 59), which also guarantee that the closure of the image is the image of the closure by a linear map. However, for the purposes of this dissertation, the statement of theorem 2.6 is enough.

Below we present an application of Theorem 2.6 to the particular case where A is the sum operator. This result will be used to establish the conditions for the closure of the sum of cone lifts K_i to be equal to the sum of closures $\mathrm{cl}\,K_i$.

Corollary 2.7 (Closure of sum of convex sets)**.** *Let $D_1, D_2, \dots, D_m$ be nonempty convex sets. Suppose that for all choices of recession directions $d_i \in \operatorname{recc}(\operatorname{cl}(D_i))$ such that $d_1 + \cdots + d_m = 0$, their opposite directions $-d_i$ also belong to $\operatorname{recc}(\operatorname{cl}(D_i))$, for all $i = 1, \dots, m$. Then,*

$$\operatorname{cl}(D_1 + \cdots + D_m) = \operatorname{cl}(D_1) + \cdots \operatorname{cl}(D_m),$$
$$\operatorname{recc}(\operatorname{cl}(D_1 + \cdots + D_m)) = \operatorname{recc}(\operatorname{cl}(D_1)) + \cdots + \operatorname{recc}(\operatorname{cl}(D_m)).$$

In particular, if each D_i is closed then $D_1 + \cdots + D_m$ is also closed.

Proof. Let $\widetilde{D}$ be the set $D_1 \times \cdots \times D_m$, and let A be the addition map, that is, $A(x_1, \dots, x_m) = x_1 + \cdots + x_m$. Note that $\widetilde{D}$ is a convex set, and the closure of $\widetilde{D}$ is equal to $\operatorname{cl}(D_1) \times \cdots \times \operatorname{cl}(D_m)$. In particular,

$$\operatorname{recc}(\operatorname{cl} \widetilde{D}) = \operatorname{recc}(\operatorname{cl} D_1) \times \cdots \times \operatorname{recc}(\operatorname{cl} D_m).$$

Thus, the intersection between the kernel of A and the recession cone of $\operatorname{cl}(\widetilde{D})$ has the following form:

$$\left\{ (d_1, \dots, d_m) \in \mathbb{R}^{mn} \;\middle|\; \begin{array}{c} d_1 + \cdots + d_m = 0 \\ d_i \in \operatorname{recc}(\operatorname{cl}(\widetilde{D}_i)),\ i = 1, \dots, m. \end{array} \right\},$$

which by hypothesis is contained in $-\operatorname{recc}(\operatorname{cl}(\widetilde{D}))$. Using Theorem 2.6 (page 21), we conclude the proof. □

Although the image by a linear map of a closed convex set may not be closed, the finite description of polyhedra given by Minkowski-Weyl Theorem ensures that the image by any linear map of any polyhedron is always a polyhedron. In particular, it is closed.

Theorem 2.8. *Let P be a nonempty polyhedron, and let A be a linear map from $\mathbb{R}^n$ to $\mathbb{R}^l$. Then, AP is a polyhedron with recession cone equal to $A \operatorname{recc}(P)$.*

Proof. From the Minkowski-Weyl Theorem (Theorem 2.2, page 13), the polyhedron P can be given by a finite description:

$$P = \operatorname{conv}(\{v_1, \ldots, v_k\}) + \operatorname{cone}(\{w_1, \ldots, w_l\}).$$

Since both the convex hull and conic hull are obtained from finite sets, we have

$$\begin{aligned} AP &= A \operatorname{conv}(\{v_1, \ldots, v_k\}) + A \operatorname{cone}(\{w_1, \ldots, w_l\}) \\ &= \operatorname{conv}(\{Av_1, \ldots, Av_k\}) + \operatorname{cone}(\{Aw_1, \ldots, Aw_l\}). \end{aligned}$$

Again by Minkowiski-Weyl Theorem, the set AP is a polyhedron and its recession cone is equal to

$$\operatorname{recc}(AP) = \operatorname{cone}(\{Aw_1, \ldots, Aw_l\}) = A \operatorname{recc}(P). \qquad \square$$

From Theorem 2.8, we obtain the following characterization for projections of polyhedra that will be useful in chapter 5 about Disjunctive Constraints:

Corollary 2.9 (Projection of polyhedron)**.** *Let P be a projection of a nonempty polyhedron:*

$$P = \{x \in \mathbb{R}^n \mid Ax + By \leq b,\ y \in \mathbb{R}^m\}$$

Then, P is also a polyhedron and the corresponding recession cone is given by

$$\operatorname{recc}(P) = \{d_x \in \mathbb{R}^n \mid Ad_x + Bd_y \leq 0,\ d_y \in \mathbb{R}^m\}.$$

Proof. Let Q be the projected polyhedron,

$$Q = \{(x, y) \in \mathbb{R}^{m+n} \mid Ax + By \leq b\},$$

and let $A \in \mathbb{R}^{n\times(m+n)}$ be the projection map, $A = \begin{bmatrix} I & 0 \end{bmatrix}$. The proof follows from Theorem 2.8 (page 25) applied to Q and A. $\square$

2.4 Convex hull of union of closed convex sets

In this section, we will prove the main result (2.1) in the context of union of closed convex sets and also union of polyhedra. We note that the case of the union of closed convex sets requires the main results of each previous sections, whereas for union of polyhedra, only the Minkowski-Weyl theorem is required. A result equivalent to the Theorem 2.10, but with a different statement, is proved in ROCKAFELLAR [22] page 80 (Theorem 9.8). Figure 2.7 illustrates the statement of Theorem 2.10.

Theorem 2.10 (Closure of convex hull of finite union)**.** *Let $D_1, \ldots, D_m$ be nonempty closed convex sets. Then,*

$$\operatorname{conv}\left(\bigcup_{i=1}^{m} D_i\right) + \sum_{i=1}^{m} \operatorname{recc}(D_i) \subseteq \operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{m} D_i\right). \tag{2.2}$$

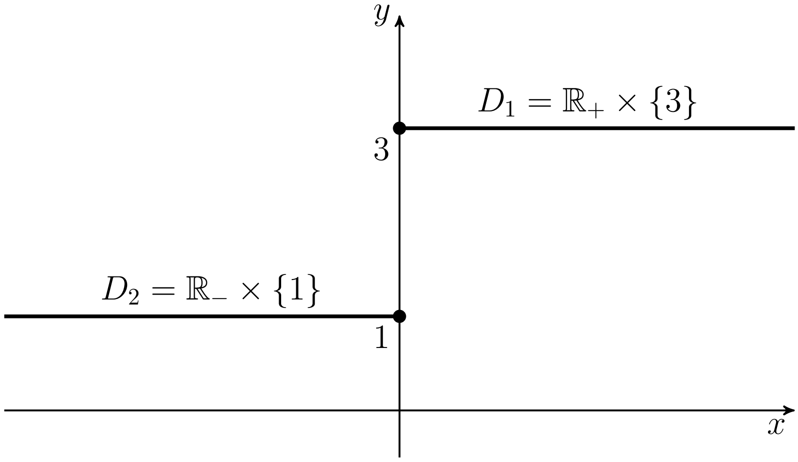
Additionally, if for all choice of directions $d_i \in \operatorname{recc}(D_i)$ which sum to zero, $d_1 + \cdots + d_m = 0$, their opposite directions $-d_i$ also belong to $\operatorname{recc}(D_i)$, then we have equality in (2.2), and the recession cone of $\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{m} D_i\right)$ is equal to $\sum_{i=1}^{m} \operatorname{recc}(D_i)$.

Proof. Let D be the union of closed convex sets $\cup_{i=1}^{m} D_i$, and let $\widetilde{D}$ be the convex hull of D plus the sum of recession cones $\sum_{i=1}^{m} \operatorname{recc}(D_i)$:

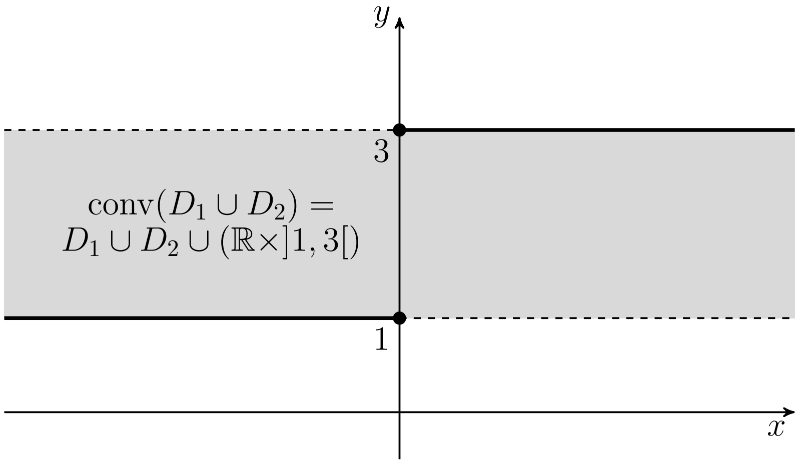
$$\widetilde{D} = \operatorname{conv}(D) + \sum_{i=1}^{m} \operatorname{recc}(D_i).$$

Let x be a vector of $\widetilde{D}$. Then, there are $x_i \in D_i$, $d_i \in \operatorname{recc}(D_i)$, and $\lambda_i \in [0,1]$, for $i = 1, \ldots, m$, such that $\sum_{j=1}^{m} \lambda_j = 1$ and

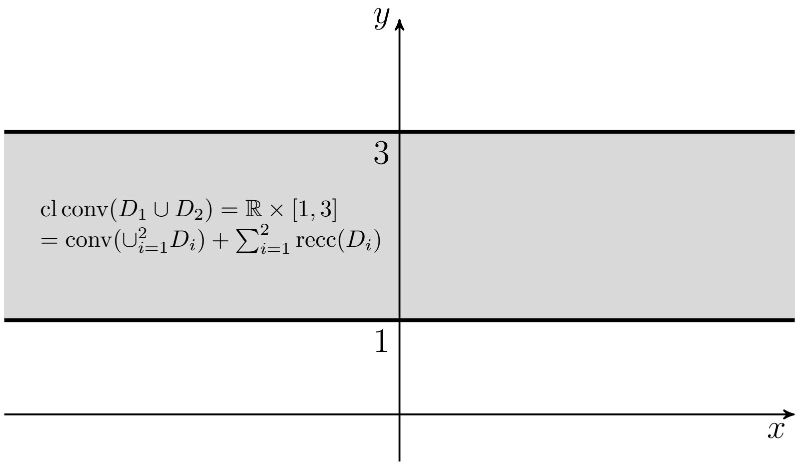
$$x = \sum_{i=1}^{m} \lambda_i x_i + \sum_{i=1}^{m} d_i.$$



(a) $D_1 \cup D_2$.



(b) $\mathrm{conv}(D_1 \cup D_2)$



(c) $\mathrm{cl}\,\mathrm{conv}(D_1 \cup D_2) = \mathrm{conv}(D_1 \cup D_2) + \sum_{i=1}^2 \mathrm{recc}(D_i)$.

Figure 2.7: Convex hull of union of closed convex sets.

For each $1 \leq i \leq 2m$, let $\{\alpha_i^k\}_{k\in\mathbb{N}}$ be a sequence in $[0,1]$ such that $\sum_{i=1}^{2m} \alpha_i^k$ is equal to 1, for all $k \in \mathbb{N}$, and the limit of α_i^k when k goes to infinity is equal to

$$\lim_{k\to\infty} \alpha_i^k = \begin{cases} \lambda_i^- & \text{, if } 1 \leq i \leq m, \\ 0^+ & \text{, if } m+1 \leq i \leq 2m. \end{cases}$$

For instance, we can consider the sequence $\{\alpha_i^k\}_{k\in\mathbb{N}}$ given by

$$\alpha_i^k = \begin{cases} \lambda_i - \frac{r}{k+1} & \text{, if } 1 \leq i \leq m \text{ and } \lambda_i > 0, \\ 0 & \text{, if } 1 \leq i \leq m \text{ and } \lambda_i = 0, \\ \frac{r}{k+1} & \text{, if } m+1 \leq i \leq m+p, \\ 0 & \text{, if } m+p+1 \leq i \leq 2m, \end{cases}$$

where r is the lowest value of λ_i among all λ_i greater than 0, that is, $r = \min_{\lambda_i>0} \lambda_i$, and p is the number of λ_i greater than 0. Let $\{y_k\}_{k\in\mathbb{N}}$ be the sequence given by

$$y_k = \sum_{i=1}^{m} \alpha_i^k x_i + \sum_{i=1}^{m} \alpha_{i+m}^k \left(x_i + \frac{1}{\alpha_{i+m}^k} d_i \right),$$

and note that y_k belongs to $\operatorname{conv}(D)$, for every $k \in \mathbb{N}$. Taking the limit in k, we have that y_k converges to x. Therefore, x belongs to $\operatorname{cl}(\operatorname{conv}(D))$.

We now proceed to the equality case in (2.2) under the additional regularity condition on the recession cones. Let K_i be the conical lift of D_i, that is, $K_i = \operatorname{cone}(\{1\} \times D_i)$, and let K be the product of cones $K_1 \times \cdots \times K_m$. Let A be the addition map

$$A(v_1, \ldots, v_m) = v_1 + \cdots + v_m,$$

where $v_i \in \mathbb{R}^{n+1}$, for all $i = 1, \ldots, m$. From Corollary 2.5 page 18

$$\begin{aligned} AK &= \operatorname{cone}\big(\{1\} \times \operatorname{conv}(D)\big), \\ A\operatorname{cl}(K) &= \operatorname{cone}\big(\{1\} \times (\operatorname{conv}(D) + R)\big) \cup \big(\{0\} \times R\big), \end{aligned}$$

where $R = \sum_{i=1}^{m} \operatorname{recc}(D_i)$, and from Theorem 2.4 page 16 applied to AK, we get:

$$\operatorname{cl}(AK) = \operatorname{cone}\big(\{1\} \times \operatorname{cl}(\operatorname{conv}(D))\big) \cup \big(\{0\} \times \operatorname{recc}(\operatorname{cl}(\operatorname{conv}(D)))\big).$$

So, it is sufficient to prove that $\text{cl}(AK)$ is equal to $A\,\text{cl}(K)$. For this purpose, we want to apply Theorem 2.6 (page 21) about the interchangeability between the linear map and closure. Recall that the recession cone of a convex cone is the cone itself. Therefore, we must verify if the intersection between the kernel of A and $\text{cl}(K)$ is contained in $-\text{cl}(K)$. Let v be a vector of $\ker(A) \cap \text{cl}(K)$. Then, v is equal to $(v_1, \ldots, v_m)$, for some $v_1 \in \text{cl}(K_1), \ldots, v_m \in \text{cl}(K_m)$, and $v_1 + \cdots + v_m = 0$. Since each v_i is equal to (λ_i, u_i) for some $\lambda_i \geq 0$ and $u_i \in \mathbb{R}^n$, we have that

$$\lambda_1 + \cdots + \lambda_m = 0, \quad \text{and} \quad u_1 + \cdots + u_m = 0.$$

Therefore, each λ_i is equal to 0, and u_i belongs to $\text{recc}(D_i)$, by the characterization of $\text{cl}(K_i)$, for all $i = 1, \ldots, m$. By hypothesis, $-u_i$ belongs to $\text{recc}(D_i)$, and so $-v$ belongs to $\text{cl}(K)$. □

One article that motivated this chapter was CERIA; SOARES [12], where the authors extend the theory of Disjunctive Programming to closed convex sets. There, the authors establish the fundamental formula (2.1) for a class of closed convex set that are included in the statement of Theorem 2.10, namely lower or upper bounded sets. Indeed, in this case the recession directions always have positive (respectively negative) components, so their sum cannot vanish, except for being all equal to zero. Then, they claim that the same result holds for the union of arbitrary closed convex sets. Figure 2.8 presents a counterexample to this statement, violating the regularity condition for sums of recession directions: $d_1 + d_2 = (1, 0) + (-1, 0) = 0$, but $-d_1$ is not in $\text{recc}(D_1)$.

An interesting result is that if all closed convex sets have a common recession cone, then the convex closure of the union also closed.

Corollary 2.11 (Common recession cone). *Let $D_1, \ldots, D_m$ be nonempty closed con-*

vex sets such that $\operatorname{recc}(D_i) = R$, *for all* $i = 1, \ldots, m$. *Then*

$$\operatorname{conv}\left(\bigcup_{i=1}^{m} D_i\right) \text{ is closed.}$$

and its recession cone is also R.

Proof. Denote by $\widetilde{D}$ the convex hull of $\bigcup_{i=1}^{m} D_i$. From Theorem 2.10 (page 27), we just have to prove that

$$\widetilde{D} + R = \widetilde{D},$$

since $\widetilde{D} + R$ is equal to $\operatorname{cl}(\widetilde{D})$. Note that $\widetilde{D}$ is contained in $\widetilde{D} + R$, because the vector 0 is also a recession direction. Let x be a vector of $\widetilde{D} + R$. Then, for each i, there is $x_i \in D_i$, $d \in R$, and $\lambda_i \in [0, 1]$ such that $\sum_{i=1}^{m} \lambda_i = 1$ and

$$x = \sum_{i=1}^{m} \lambda_i x_i + d = \sum_{i=1}^{m} \lambda_i (x_i + d).$$

Since $x_i + d$ belongs to D_i for each i, x belongs to $\widetilde{D}$. □

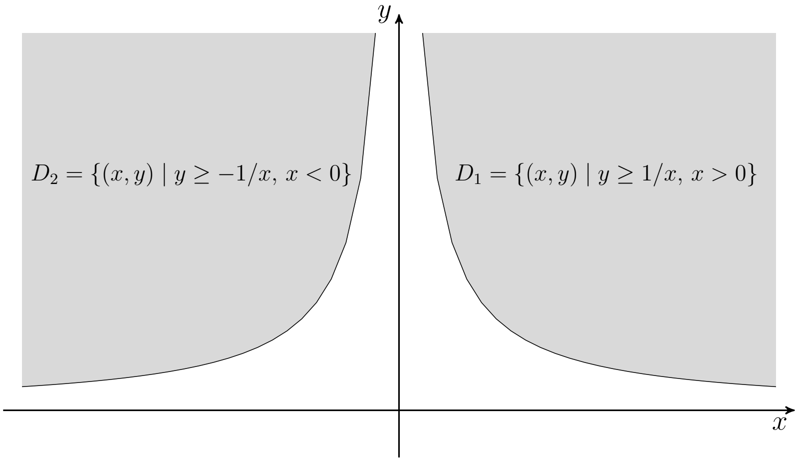
Curiously, if we don't have equality in formula (2.1) (page 9, then the convex closure $\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{m} D_i\right)$ has no extreme points. This will appear in the proof of the Blessing of Extreme Points in chapter 7.

Corollary 2.12. *(Lack of extreme points) Let* $D_1, \ldots, D_m$ *be nonempty closed convex sets. If we have strict inclusion in* (2.2)*, that is, if*

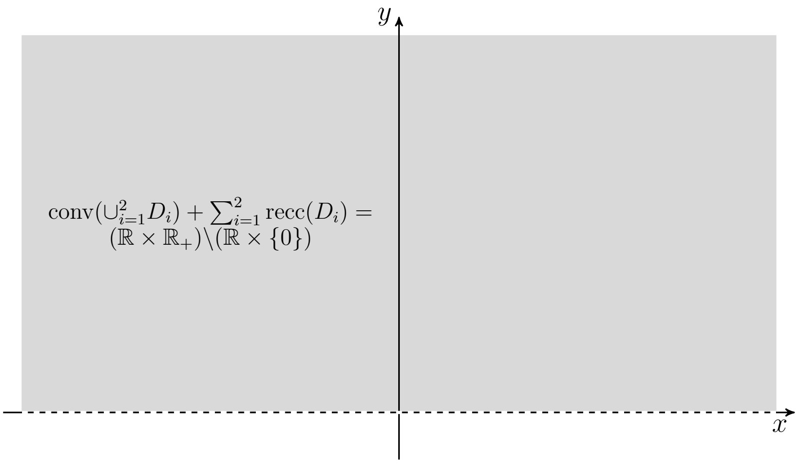
$$\operatorname{conv}\left(\bigcup_{i=1}^{m} D_i\right) + \sum_{i=1}^{m} \operatorname{recc}(D_i) \subsetneq \operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{m} D_i\right), \tag{2.3}$$

then $\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{m} D_i\right)$ *has no extreme points.*

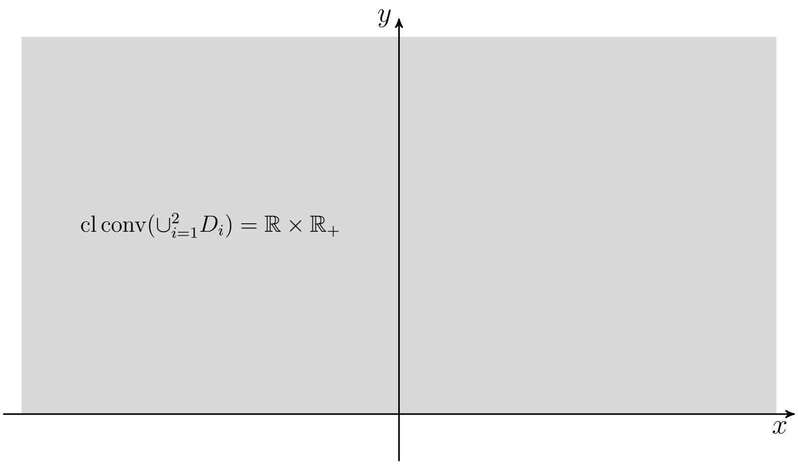
Proof. If condition (2.3) holds, then there are recession directions $d_i \in \operatorname{recc}(D_i)$ such that $d_1 + \cdots d_m = 0$, but an opposite direction $-d_j$ does not belongs to $\operatorname{recc}(D_j)$,



(a) $D_1 \cup D_2$.



(b) $\operatorname{conv}(D_1 \cup D_2)$.



(c) $\operatorname{cl}\operatorname{conv}(D_1 \cup D_2)$.

Figure 2.8: Lack of extremes points when the convex hull formula (2.2) does not holds as an equality.

for some j. Note that d_j is not equal to 0, since the vector 0 belongs to every cone. Moreover, we have that

$$-d_j \in \sum_{\substack{i=1 \\ i \neq j}}^{m} \operatorname{recc}(D_i).$$

Therefore, both vectors d_j and $-d_j$ belong to the recession cone of $\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{m} D_i\right)$, and so $\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{m} D_i\right)$ has no extreme points. $\square$

As in other cases, the same result for the polyhedral case always hold without any additional regularity condition.

Theorem 2.13 (Closure of convex hull of union of polyhedra)**.** *Let $P_1, \ldots, P_m$ be nonempty polyhedra. Then,*

$$\operatorname{conv}\left(\bigcup_{i=1}^{m} P_i\right) + \sum_{i=1}^{m} \operatorname{recc}(P_i) = \operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{m} P_i\right). \tag{2.4}$$

In particular, the set $\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{m} P_i\right)$ is a polyhedron with recession cone equal to $\sum_{i=1}^{m} \operatorname{recc}(P_i)$.

Proof. Denote by $\widetilde{P}$ the set $[\operatorname{conv}\left(\cup_{i=1}^{m} P_i\right) + \sum_{i=1}^{m} \operatorname{recc}(P_i)]$. From Theorem 2.10 (page 27), we have the following inclusion

$$\operatorname{conv}\left(\bigcup_{i=1}^{m} P_i\right) \subseteq \widetilde{P} \subseteq \operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{m} P_i\right),$$

so it is sufficient to prove that $\widetilde{P}$ is a closed set.

According to the Minkowiski-Weyl Theorem, each polyhedron P_i can be represented by a convex combination plus a positive combination of a fixed number of vectors, denoted by $\{v_{i,r}\}_{r=1}^{k_i}$ and $\{w_{i,s}\}_{s=1}^{l_i}$, respectively. We claim that

$$\widetilde{P} = \operatorname{conv}\left(\bigcup_{i=1}^{m} \{v_{i,r}\}_{r=1}^{k_i}\right) + \operatorname{cone}\left(\bigcup_{i=1}^{m} \{w_{i,s}\}_{s=1}^{l_i}\right).$$

Indeed, let Q be the polyhedron of the right-hand side. Note that Q is a subset of $\widetilde{P}$, since

$$\bigcup_{i=1}^{m} \{v_{i,r}\}_{r=1}^{k_i} \subset \operatorname{conv}\left(\bigcup_{i=1}^{m} P_i\right),$$
$$\bigcup_{i=1}^{m} \{w_{i,s}\}_{s=1}^{l_i} \subset \sum_{i=1}^{m} \operatorname{recc}(P_i),$$

and the sets on the right-hand side are already convex conical, respectively. We shall prove the opposite inclusion. Let $\widetilde{x}$ be a vector of $\widetilde{P}$. Then, x has the form

$$\begin{aligned} x &= \sum_{i=1}^{m} \lambda_i x_i + \sum_{i=1}^{m} \alpha_i d_i \\ &= \sum_{i=1}^{m} \left(\sum_{r=1}^{k_i} \widetilde{\lambda}_{i,r} v_{i,r} \right) + \sum_{i=1}^{m} \left(\sum_{s=1}^{l_i} \widetilde{\alpha}_{i,s} w_{i,s} \right). \end{aligned}$$

for some vectors $x_i \in P_i$, directions $d_i \in \operatorname{recc}(P_i)$ and scalars $\lambda_i, \widetilde{\lambda}_{i,r}, \alpha_i, \widetilde{\alpha}_{i,s} \geq 0$, such that $\sum_j \lambda_j = \sum_{j,r} \widetilde{\lambda}_{j,r} = 1$. Therefore, x belongs to Q. □

A similar result to Corollary 2.11 is also valid in the polyhedral case.

Corollary 2.14 (Common recession cone of polyhedra)**.** *Let $P_1, \ldots, P_m$ be nonempty polyhedra with a common recession cone,* $\operatorname{recc}(P_i) = R$*, for all $i = 1, \ldots, m$. Then,*

$$\operatorname{conv}\left(\bigcup_{i=1}^{m} P_i\right) \quad \textit{is a polyhedron,}$$

and the corresponding recession cone is R.

Proof. From Corollary 2.11 (page 30), the set $\operatorname{conv}(\cup_{i=1}^{m} P_i)$ is closed with recession cone equal to R, and from Theorem 2.13 (page 33) we have that $\operatorname{cl}\operatorname{conv}(\cup_{i=1}^{m} P_i)$ is a polyhedron. □

3 OPTIMAL VALUE FUNCTIONS

3.1 More basic results on convex analysis

The ideas throughout this dissertation are described in terms of *extended real valued functions*, i.e., functions that can take values $-\infty$ and $+\infty$ at some points. In convex optimization, the extended real valued functions bring in important geometric insights that connect results from convex sets with those from convex functions. They also provide a unified framework to analyze convex optimization problems, especially when the objective function may assume infinite values, as in the case of objective functions that are the pointwise supremum of a family of functions, e.g., the cost-to-go functions from a dynamic programming problem.

Traditional constrained optimization problems can be easily transformed into unconstrained ones by using extended real value functions. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function, and let X be a subset of $\mathbb{R}^n$, and consider the problem of minimizing f over X:

$$\begin{array}{rl} \min_x & f(x) \\ \text{s.t.} & x \in X. \end{array}$$

Let us denote by δ_X the *indicator function* of X which is equal to 0 if x belongs to X, or equal to $+\infty$ otherwise:

$$\delta_X(x) = \begin{cases} 0 & \text{if } x \in X, \\ +\infty & \text{otherwise.} \end{cases}$$

Then, the constrained problem is equivalent to minimizing $f + \delta_X$ over $\mathbb{R}^n$. Using this idea, we can *always* restrict our attention to the unconstrained minimization of extended real valued functions. Next, we introduce some concepts that establish the connections between extended real valued functions and sets.

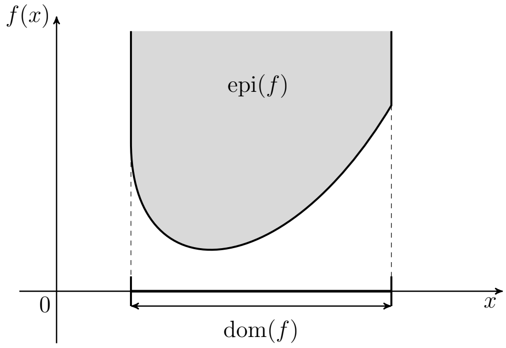
Let X be a subset of $\mathbb{R}^n$. The *epigraph* of a function $f : X \to [-\infty, \infty]$ is the set of points lying on or above the graph of f:

$$\mathrm{epi}(f) = \{(x, w) \in \mathbb{R}^{n+1} \mid x \in X,\ f(x) \leq w\}.$$

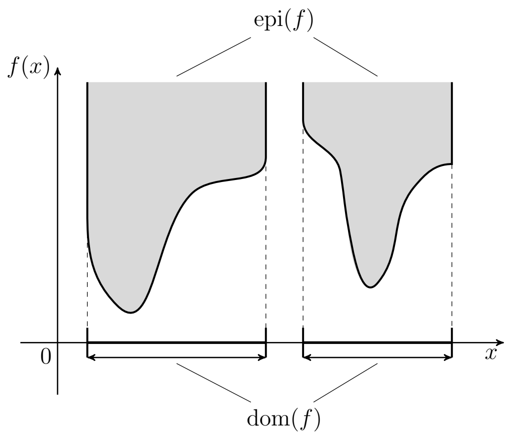
The *effective domain* of f is the set of points whose image by f is less than $+\infty$:

$$\mathrm{dom}(f) = \{x \in X \mid f(x) < \infty\},$$

see figure 3.1. It is instructive to note that $\mathrm{dom}(f)$ is the projection of $\mathrm{epi}(f)$ on $\mathbb{R}^n$, that is, $\mathrm{dom}(f)$ is equal to $\mathrm{proj}_x(\mathrm{epi}(f))$. Since we are dealing with a general setting, it is important to exclude some degenerate cases of extended real-valued functions. We say that f is *proper* if there is a point $x \in X$ such that $f(x) < \infty$ (the epigraph of f is non-empty), and $f(y) > -\infty$ for all $y \in X$ (the epigraph of f does not contain a vertical line). The properness assumption is crucial for most results we give.



(a) Convex function.



(b) Non-convex function.

Figure 3.1: Convex (left) and non-convex function (right).

Another important concept in optimization is the definition of a convex function. A difficulty for defining convexity for extended real-valued functions is that they may assume both values $-\infty$ and $+\infty$, so the standard definition

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$$

may result in undefined expression $+\infty - \infty$. The epigraph provides an effective and geometric way of dealing with this difficulty. We say that $f : \mathbb{R}^n \to [-\infty, \infty]$ is *convex* if $\text{epi}(f)$ is a convex subset of $\mathbb{R}^{n+1}$, see again figure 3.1. It is a classical result to show the following properties of operations on convex functions:

- Let $f : \mathbb{R}^n \to [-\infty, \infty]$ be a convex function. The function αf is convex, for any non-negative scalar $\alpha \geq 0$;
- Let $f, g : \mathbb{R}^n \to (-\infty, +\infty]$ be two convex functions. The sum $f + g$ is well defined and convex;
- Let $f_i : \mathbb{R}^n \to [-\infty, \infty]$ be convex functions, for all $i \in I$. Then, the pointwise supremum $\sup_{i \in I} f_i(x)$ is also a convex function;
- Let D be a convex set. Then, the indicator function δ_D is a convex function.

So, we say that the four operations, respectively *positive scaling*, *sums*, *pointwise supremum* and *indicator* are "convex-preserving operations".

It is also worth mentioning that the level set of a convex function f,

$$L_\alpha = \{x \in \mathbb{R}^n \mid f(x) \leq \alpha\},$$

is a convex set for every $\alpha \in \mathbb{R}$, since it is the projection in $\mathbb{R}^n$ of the convex set $\text{epi}(f) \cap \big(\mathbb{R}^n \times (-\infty, \alpha]\big)$.

We say that a function $f : X \to [-\infty, \infty]$ is *closed* if the epigraph of f is a closed set, see figure 3.2. The closedness of a function is related to the notion of

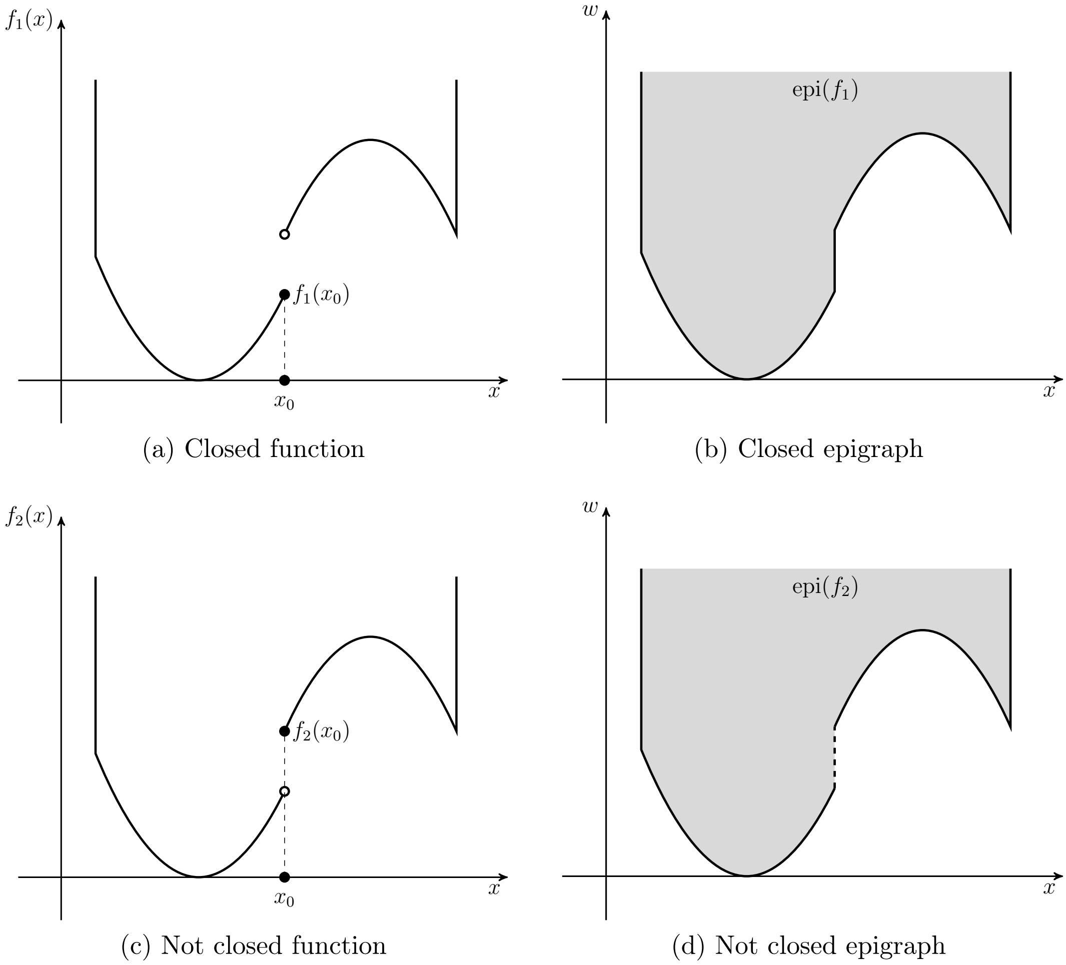


Figure 3.2: Closed and non-closed functions.

lower semicontinuity. A function f is *lower semicontinuous* at a point $x \in X$ if

$$f(x) \leq \liminf_{k \to \infty} f(x_k),$$

for every sequence $\{x_k\} \subset X$ with $x_k \to x$. We say that f is *lower semicontinuous* if it is lower semicontinuous at every point x in its domain X. The following theorem connects the notions of closedness, lower semicontinuity and closedness of level sets for extended real valued functions. For a proof, see ROCKAFELLAR [22], page 51 or BERTSEKAS [5], page 10.

Theorem 3.1 (Characterization of closed functions)**.** *Let f be an arbitrary function from $\mathbb{R}^n$ to $[-\infty, \infty]$. Then the following conditions are equivalent:*

a) f is lower semicontinuous throughout $\mathbb{R}^n$;

b) $\{x \in \mathbb{R}^n \mid f(x) \leq \alpha\}$ is closed for every $\alpha \in \mathbb{R}$;

c) The epigraph of f is a closed set in $\mathbb{R}^{n+1}$.

From this theorem, it is also not hard to show that the same 4 operations that preserve convexity, also preserve closedness: for example, the pointwise supremum f of closed functions f_k is closed, since the epigraph $\text{epi}(f)$ is the intersection of closed epigraphs $\text{epi}(f_k)$, and therefore also closed. The same holds for positive scaling, sums and the construction of indicator functions.

Another consequence, is that if f is a closed proper convex function, then the epigraph $\text{epi}(f)$ is a nonempty closed convex set.

As seen in section 2, an important concept from closed convex sets is the recession cone. We also have an analogous notion for proper closed convex functions that plays a central role in convex optimization. The *recession cone* of f is the set of "horizontal directions" from the recession cone of the epigraph of f:

$$\text{recc}(f) = \{d \in \mathbb{R}^n \mid (d, 0) \in \text{recc}(\text{epi}(f))\}.$$

In other words, a recession direction of a closed proper convex function f is a direction from $\mathbb{R}^n$ along which the function f "recedes", that is, does not grow to infinity. The following theorem connects the notion of recession cone of a nonempty level set and the recession cone of the corresponding proper closed convex function.

Theorem 3.2. *Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a closed proper convex function and consider the level set*

$$V_\gamma = \{x \in \mathbb{R}^n \mid f(x) \leq \gamma\}, \quad \gamma \in \mathbb{R}.$$

Then, all the nonempty level sets V_γ have the same recession cone:

$$\mathrm{recc}(V_\gamma) = \mathrm{recc}(f),$$

for all γ greater than $\inf_{x \in \mathbb{R}^n} f(x)$.

The notion of recession directions of closed proper functions is an important concept for analyzing existence of optimal solutions and optimal value functions of convex programs, as we will see in Theorems 3.4 (page 44) and Theorem 3.11 (page 53). In particular, these results establish some sufficient conditions for the optimal value function f of a special type of convex optimization problem to be a closed proper convex function, namely for

$$\begin{aligned} f(b) = \min \quad & F(x) \\ \text{s.t.} \quad & g(x) \leq b, \end{aligned}$$

where F and g are closed proper convex functions. This is key to understanding the properties of the cost-to-go functions of a dynamic programming problem.

A particular class of extended real-valued functions that arise naturally as the optimal value function from a linear program is the class of polyhedral functions. We say that f is a *polyhedral function* if the epigraph of f is a nonempty polyhedron with no negative vertical directions, see figure 3.3. In particular, the function f is closed, proper and convex. We have, again, the same 4 operations preserving polyhedral functions. However, some care must be taken: since we must preserve its finite description, we don't allow infinite index sets for supremum, and we also impose regularity conditions so that the function does not become identically infinite:

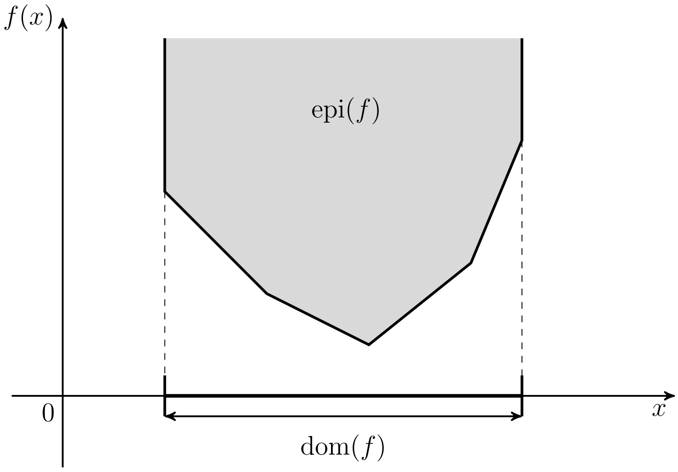


Figure 3.3: Polyhedral function.

- Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a polyhedral function. The function αf is polyhedral, for any non-negative scalar $\alpha \geq 0$;

- Let $f, g : \mathbb{R}^n \to (-\infty, +\infty]$ be two polyhedral functions;. The sum $f + g$ is polyhedral, if $\mathrm{dom}(f) \cap \mathrm{dom}(g) \neq \emptyset$ (nonempty epigraph);

- Let $f_i : \mathbb{R}^n \to (-\infty, \infty]$ be a polyhedral function, for all $i \in I$, and suppose that I is finite. Then, the pointwise supremum $\sup_{i \in I} f_i(x)$ is a polyhedral function, if $\cap_{i \in I} \mathrm{dom}(f_i) \neq \emptyset$ (nonempty epigraph);

- Let P be a polyhedron. Then, the indicator function δ_P is a polyhedral function.

The following theorem provides a useful representation of polyhedral functions, and the proof can be found on page 110 from BERTSEKAS [5].

Theorem 3.3. *The function $f : \mathbb{R}^n \to (-\infty, \infty]$ is polyhedral if and only if it has the form*

$$f(x) = \begin{cases} \max_{j=1,\ldots,m}\{a_j^\top x + b_j\} & , \textit{if } Wx \leq h, \\ +\infty & , \textit{otherwise,} \end{cases}$$

where a_j are vectors in $\mathbb{R}^n$, b_j are scalars, m is a positive integer, W is a matrix, and h is a vector such that $\mathrm{dom}(f) = \{x \in \mathbb{R}^n \mid Wx \leq h\}$.

An example of polyhedral function is the linear function $f(x) = c^\top x$. We shall see in Corollary 3.12 that the optimal value function of an optimization problem with polyhedral objective function and linear constraints is also polyhedral. In particular, the optimal value of a linear programming (LP) problem

$$\begin{aligned} f(b) = \min \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b, \\ & x \in \mathbb{R}^n, \end{aligned}$$

is also polyhedral. Those are important results for understanding properties of the cost-to-go functions in the dynamic programming problem. In the general case, the infimum operation requires additional regularity conditions to preserve convexity and closedness. We present those conditions in section 3.3,.

One last idea we will explore is the union of polyhedra which establishes a connection between convex programming techniques and non-convex problems such as mixed-integer linear programming problems (MILP). An optimal value function f of a MILP is defined by

$$\begin{aligned} f(b) = \min_{x,z} \quad & c^\top x + q^\top z \\ \text{s.t.} \quad & Ax + Gz \leq b, \\ & (x, z) \in \mathbb{R}^n \times \mathbb{Z}^l, \end{aligned} \tag{3.1}$$

where the vectors and matrices have appropriate dimensions. Using Benders decomposition BENDERS [4], we can represent f by the minimum of polyhedral functions f_z, $f(b) = \min_{z \in \mathbb{Z}^l} f_z(b)$, where each $f_z(b)$ is the optimal value function of a LP plus a linear function:

$$\begin{aligned} f_z(b) = q^\top z + \min_{x} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b - Gz, \\ & x \in \mathbb{R}^n. \end{aligned}$$

From a geometric perspective, the epigraph of f is described by the union of the epigraphs of all f_z:

$$\begin{aligned}\operatorname{epi}(f) &= \{(b,w) \in \mathbb{R}^{m+1} \mid f(b) \leq w\} \\ &= \{(b,w) \in \mathbb{R}^{m+1} \mid f_z(b) \leq w, \text{ for some } z \in \mathbb{Z}^l\} \\ &= \bigcup_{z \in \mathbb{Z}^l} \operatorname{epi}(f_z).\end{aligned}$$

For simplicity, we restrict the analysis of optimal value functions of a MILP to problems in which the integer variable can only assume a finite number of different values. In particular, we will focus on 0-1 MILP problems where the variable z from (3.1) belongs to $\{0,1\}^l$, instead of $\mathbb{Z}^l$. Indeed, these MILP value functions belong to the class of *piecewise polyhedral function*, which are the extended real-valued functions $f : \mathbb{R}^n \to (\infty, \infty]$ whose epigraph is a finite union of polyhedra. Piecewise polyhedral functions are also preserved by the same 4 operations of polyhedral functions: positive scaling, sums of functions with intersecting domains, finite maximum and construction of indicator function. This last one is:

- Let P be a union of polyhedra. Then, the indicator function δ_P is a piecewise polyhedral function.

We note that not every piecewise polyhedral function is an optimal value function from a 0-1 MILP problem. The reason for that is related to Theorem 5.2 (page 98) of chapter 5.

3.2 Existence of primal solutions

In this section we focus on regularity conditions for the existence of optimal solutions. In convex optimization, even if the objective function is bounded below or the corresponding infimum is finite, there is no guarantee that the minimum will be achieved by a feasible solution. We present some sufficient conditions for the existence of optimal solutions which are closely related to the idea of recessions directions.

From this section onwards, we assume that the optimization problems have finite optimal value. In applications, this is a reasonable hypothesis since the capacity to improve a system, reduce cost, or increase profit is usually limited. We seize this occasion to present analogous results for MILPs, building upon the polyhedral case.

We introduce the notion of lineality space of a set X and a function f for the statement of the next result. The lineality space of a set X is $\operatorname{recc}(X) \cap (-\operatorname{recc}(X))$, and the lineality space of a function f is given by $\operatorname{recc}(f) \cap (-\operatorname{recc}(f))$.

Theorem 3.4 (Existence of primal solution of convex problems). *Let X be a closed convex set, and let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a closed convex function with $X \cap \operatorname{dom}(f) \neq \emptyset$. Assume that the optimal value of the problem*

$$\begin{array}{rl} \min_x & f(x) \\ s.t. & x \in X \end{array}$$

is finite, and the recession cones and lineality space of both X and f satisfy the condition

$$\operatorname{recc}(X) \cap \operatorname{recc}(f) = \operatorname{lineal}(X) \cap \operatorname{lineal}(f).$$

Then the problem has at least one optimal solution.

Proof. Let p^* be the optimal value. Note that $p^* \neq +\infty$, since $X \cap \operatorname{dom}(f)$ in nonempty. Let $\{\gamma_k\}$ be a monotone scalar sequence with $\gamma_k \downarrow p^*$ and denote

$$V_k = \{x \in \mathbb{R}^n \mid f(x) \leq \gamma_k\}.$$

The set $X \cap V_k$ is nonempty, by the infimum property. We claim that the intersection

$$X^* = \bigcap_{k=1}^{\infty} (X \cap V_k)$$

is nonempty. Indeed, let x_k be the optimal solution of the following problem:

$$\begin{array}{rl} \min & \|u\| \\ \text{s.t.} & u \in X \cap V_k. \end{array}$$

Since $\|\cdot\|$ has compact level sets and is strictly convex, the optimal solution x_k exists and is unique. Then, there are two possibilities for the sequence $\{x_k\}$: it is bounded, or unbounded.

If $\{x_k\}$ is bounded, there is a subsequence that converges to $\overline{x} \in \mathbb{R}^n$. Since $\{X \cap V_k\}$ is a sequence of decreasing nonempty closed sets, we have that $\overline{x}$ belongs to $X \cap V_k$, for all k. So, X^* is nonempty. If $\{x_k\}$ is unbounded, there is a subsequence $\{x_{k_j}\}$ such that

$$\lim_{j \to \infty} \frac{x_{k_j}}{\|x_{k_j}\|} = d \in \operatorname{recc}(X) \cap \operatorname{recc}(f).$$

Let y_k be the vector defined by $x_k - \|x_k\| d$, and note that y_k belongs to $X \cap V_k$, for all k, since $-d \in \operatorname{recc}(X) \cap \operatorname{recc}(f)$, by hypothesis. We note that $\{\|y_{k_j}\|/\|x_{k_j}\|\}$ converges to 0:

$$\lim_{j \to \infty} \frac{\|y_{k_j}\|}{\|x_{k_j}\|} = \lim_{j \to \infty} \left\| \frac{x_{k_j}}{\|x_{k_j}\|} - d \right\| = 0.$$

In particular, for sufficiently large j, the norm $\|y_{k_j}\|$ is smaller than $\|x_{k_j}\|$, which is a contradiction with the definition of x_k. $\square$

The polyhedral case requires less regularity conditions to obtain the same result. In an intuitive manner, the difference between both cases here is that the boundary from a polyhedron differs from a straight line just finite number of times, while the boundary from a general convex set can be a smooth curve, and so the optimal solution may diverge to infinity. For instance, let f be the function

$$f(x) = \begin{cases} 1/x, & \text{if } x > 0, \\ +\infty, & \text{if } x \leq 0. \end{cases}$$

Then, the infimum of f over $\mathbb{R}$ is equal to 0, however there is no $x^* \in \mathbb{R}$ such that $f(x^*) = 0$. Indeed, the recession cone of f is equal to $\mathbb{R}_+$, but the lineality space of f is equal to $\{0\}$. So, f does not satisfy to the statement of Theorem 3.4.

Theorem 3.5 (Existence of primal solution of polyhedral problems)**.** *Let c be a vector in $\mathbb{R}^n$, A be a matrix in $\mathbb{R}^{m\times n}$ and b be a vector in $\mathbb{R}^m$. Assume that the optimal value of the problem*

$$\begin{array}{rl} \min_x & c^\top x \\ s.t. & Ax \leq b, \\ & x \in \mathbb{R}^n \end{array}$$

is finite. Then the problem has at least one optimal solution.

The 0-1 mixed integer linear case is analogous to the polyhedral case, since we can use the Benders decomposition to analyze the minimum of a finite number of polyhedral problems. This is the main idea for

Corollary 3.6 (Existence of primal solutions of mixed integer 0–1 programs)**.** *For a standard 0-1 MILP,*

$$\begin{array}{rl} \min_{x,z} & c^\top x + q^\top z \\ s.t. & Ax + Gz \leq b, \\ & x \in \mathbb{R}^n,\ z \in \{0,1\}^l, \end{array}$$

if the optimal value is finite, then there is at least one optimal solution.

Proof. For each binary solution $z \in \{0,1\}^l$, the corresponding LP is infeasible or has finite optimal value:

$$\begin{array}{rl} \min_x & c^\top x + q^\top z \\ \text{s.t.} & Ax \leq b - Gz, \\ & x \in \mathbb{R}^n. \end{array}$$

Thus, each feasible LP has an optimal solution (Theorem 3.5). Since the original problem is feasible, there is at least one LP that has finite optimal value, and since the number of possibilities for z is finite, the infimum for the mixed 0-1 program corresponds to the lowest LP optimal value. Therefore, the optimal solution for the

0-1 MILP problem is the pair (x^*, z^*), where z^* is the binary solution associated with lowest LP optimal value and x^* is the corresponding LP optimal solution. $\square$

A very important idea in mixed integer linear programming is to approximate the convex hull of the feasible set. Under mild regularity conditions (see Theorem 3.8, page 49), the convex hull of a MILP feasible set is a polyhedron and, as we shall see below, the infimum of a linear function over a set X equals the infimum of the same linear function over the convex hull of X. From theoretical perspective, we rely on this equality to prove some results for MILPs based on results from LPs, see Corollary 3.9 (page 50).

Algorithms for solving LPs are much faster than those for solving MILPs, but an analytical expression of $\mathrm{conv}(X)$ is rarely available. In practice, many algorithms for MILPs create outer approximations of $\mathrm{conv}(X)$ using cutting planes, and use that to get lower bounds for the optimal value.

Lemma 3.7. *Let c be a vector in $\mathbb{R}^n$, and let X be a subset of $\mathbb{R}^n$, not necessarily convex. Then*

$$\inf_{x \in \mathrm{conv}(X)} c^\top x = \inf_{x \in X} c^\top x,$$

and if the infimum is attained by an element of $\mathrm{conv}(X)$ *it is also attained by some element of X.*

Proof. Let us denote by $\overline{w}$ the infimum in the left-hand side and by w the infimum in the right-hand side. Note that $\overline{w}$ is less than or equal to w, since the convex hull of X, $\mathrm{conv}(X)$, contains the original set X.

Let $\overline{x} \in \mathrm{conv}(X)$. Then, $\overline{x}$ can be described by a convex combination of a finite number of elements from X:

$$\overline{x} = \sum_{i=1}^{k} \lambda_i x_i,$$

where $x_i \in X$ and $\lambda_i \geq 0$ are such that $\sum_{i=1}^{k} \lambda_i = 1$. Taking the inner product with c, and moving $c^\top \overline{x}$ to the opposite side we get

$$\sum_{i=1}^{k} \lambda_i c^\top (x_i - \overline{x}) = 0. \tag{3.2}$$

We claim that there is some $x_j \in X$ such that $c^\top x_j \leq c^\top \overline{x}$. Indeed, if $c^\top x_i > c^\top \overline{x}$ for all $i = 1, \ldots, k$, then

$$\sum_{i=1}^{k} \lambda_i c^\top (x_i - \overline{x}) > 0,$$

which is a contradiction with (3.2). Therefore, for every $\overline{x} \in \text{conv}(X)$ there is some $x \in X$ such that $c^\top x \leq c^\top \overline{x}$. So, we conclude that $\overline{w} = w$. This also proves that the existence of optimal solutions in $\text{conv}(X)$ implies the existence of optimal solutions in X. $\square$

There are some cases that the convex hull of X is not a closed set. Let us denote by X the feasible of a MILP:

$$X = \{(x, z) \in \mathbb{R}^n \times \mathbb{Z}^l \mid Ax + Gz \leq b\}.$$

If there is an infinite number of integer feasible solutions and the matrices A or G are not rational, then the convex hull of X may not be a closed set. A classical example is the following set:

$$X = \{(x_1, x_2) \in \mathbb{Z}^2 \mid 0 \leq x_2 \leq \sqrt{2} x_1\}.$$

Note that the only point of X that belongs to the line $x_2 = \sqrt{2} x_1$ is $(0, 0)$, since $\sqrt{2}$ is irrational and the quotient x_2/x_1 is always rational. Moreover, there are rational approximations x_2/x_1 arbitrarily close to $\sqrt{2}$ such that (x_1, x_2) belongs to X, see figure 3.4a and 3.4b. If the matrices A and G were rational, then Meyer's Theorem (Theorem 3.8) states that the convex hull of the feasible set X is a polyhedron, which is a closed set. Meyer's Theorem is also known as the Fundamental Theorem of Mixed Integer Linear Programming.

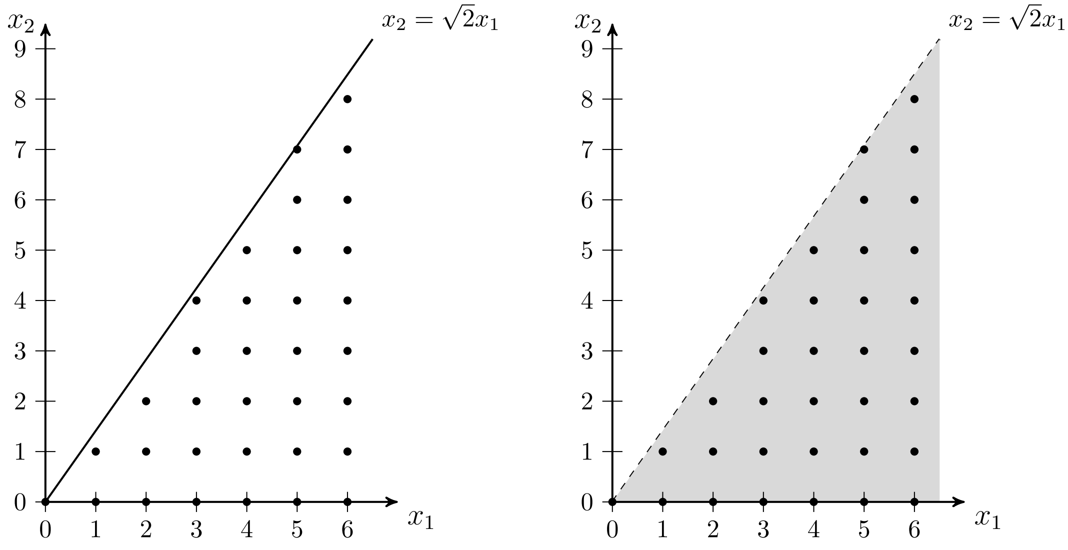


Figure 3.4: Counter-example of Meyer's Theorem when G is not rational. On the left, we have $X = \{(z_1, z_2) \in \mathbb{Z}^2 \mid 0 \leq z_2 \leq \sqrt{2}z_1\}$, and on the right, we have $\text{conv}(X)$.

Theorem 3.8 (Meyer). *Given rational matrices A, G and a vector b, let $P := \{(x, z) \in \mathbb{R}^{n+l} \mid Ax + Gz \leq b\}$ and $X := \{(x, z) \in \mathbb{R}^n \times \mathbb{Z}^l \mid (x, z) \in P\}$.*

1. *There exist rational matrices A', G' and a vector b' such that*

$$\text{conv}(X) = \{(x, z) \in \mathbb{R}^{n+l} \mid A'x + G'z \leq b'\};$$

2. *If X is nonempty, the recession cones of* $\text{conv}(X)$ *and P coincide;*

3. *If b is a rational vector, then b' is also rational.*

Now, we are in position to state and prove the theorem about existence of primal solutions of MILPs. We note that the hypothesis of rational matrices A and G is only necessary if the number of feasible integer points is infinite. For the finite case, the proof is analogous to the 0-1 MILP case.

Corollary 3.9 (Existence of primal solution of MILP problems)**.** *Given rational matrices A, G and vectors c, q, b with appropriate dimensions, consider the following mixed integer programming problem:*

$$\begin{array}{rl} \min_{x,z} & c^\top x + q^\top z \\ s.t. & Ax + Gz \leq b, \\ & x \in \mathbb{R}^n,\ z \in \mathbb{Z}^l. \end{array}$$

If the optimal value is finite, then the problem has at least one optimal solution.

Proof. We denote by X the following set

$$X = \{(x, z) \in \mathbb{R}^n \times \mathbb{Z}^l \mid Ax + Gz \leq b\}.$$

From Lemma 3.7 (page 47), the infimum over X and the infimum over $\operatorname{conv}(X)$ are equal. Therefore, the infimum over $\operatorname{conv}(X)$ is finite. By Meyer's Theorem (Theorem 3.8, page 49), the convex hull of X is a polyhedron, and by Theorem 3.5 (page 46) the following linear program

$$\begin{array}{rl} \min_{x,z} & c^\top x + q^\top z \\ \text{s.t.} & (x, z) \in \operatorname{conv}(X), \end{array}$$

has an optimal solution. From the second part of Lemma 3.7, we conclude that the original problem has an optimal solution. □

3.3 Partial minimization of functions

In this section, we analyze the properties of optimal value functions for several classes of optimization problems. With it, we setup the necessary background for the dynamic programming setting of section of section 4.1.

There is an important geometric relation between the epigraph of a given function and the epigraph of its partially minimized version: except for some boundary points, *the latter is obtained by projection from the former*, see figure 3.5. This is the key to understanding the properties of partially minimized functions.

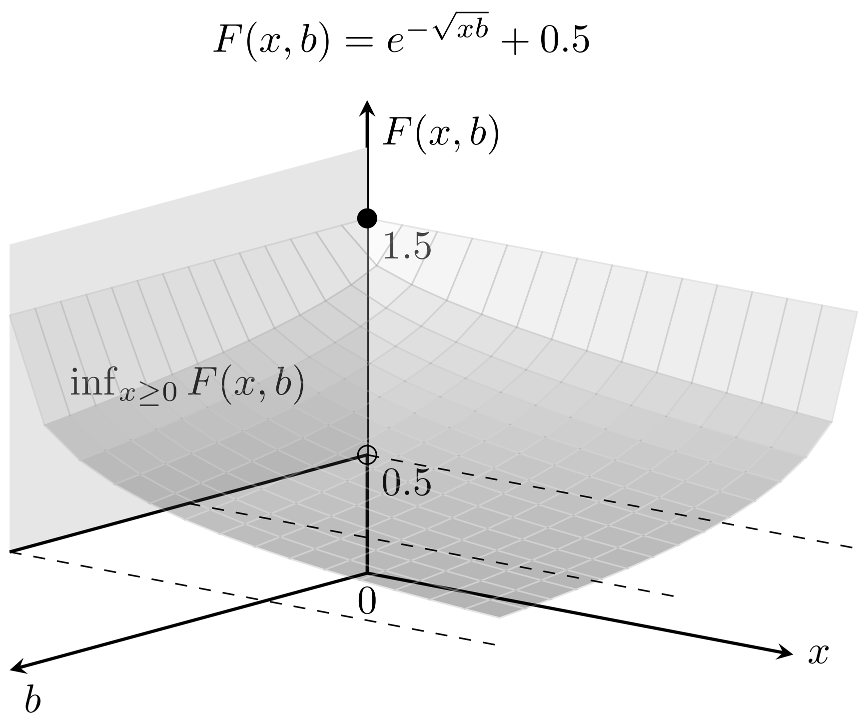


Figure 3.5: Partial minimization and epigraph projection

Theorem 3.10 (Partial minimization and epigraph projection)**.** *Consider a bounded from below proper function $F : \mathbb{R}^{n+m} \to (-\infty, \infty]$ and the function $f : \mathbb{R}^n \to [-\infty, \infty]$ defined by $f(b) = \inf_{x \in \mathbb{R}^n} F(x,b)$. Then f is proper and bounded from below with*

$$\mathrm{proj}_{(b,w)}(\mathrm{epi}\, F) \subseteq \mathrm{epi}(f) \subseteq \mathrm{cl}(\mathrm{proj}_{(b,w)}(\mathrm{epi}(F))),$$

where $\mathrm{proj}_{(b,w)}(x,b,w) = (b,w)$ *is the projection on the variables (b,w). If F is convex, then f is also convex.*

Proof. By the properness of F, the set $\mathrm{epi}(F)$ is nonempty. Let $(\overline{x}, \overline{b}, \overline{w}) \in \mathrm{epi}(F)$, and note that

$$f(\bar{b}) = \inf_{x \in \mathbb{R}^n} F(x, \bar{b}) \leq F(\bar{x}, \bar{b}) \leq \bar{w}.$$

Thus $(\bar{b}, \bar{w})$ belongs to epi(f). In particular, epi(f) is nonempty and f is a proper function, since F is bounded from below. Note that for any $(\bar{b}, \bar{w}) \in$ epi(f) and every k, there is $\bar{x} \in \mathbb{R}^n$ such that

$$\inf_{x \in \mathbb{R}^n} F(x, \bar{b}) \leq F(\bar{x}, \bar{b}) \leq \inf_{x \in \mathbb{R}^n} F(x, \bar{b}) + \frac{1}{k} \leq \bar{w} + \frac{1}{k}.$$

Then, we have $(\bar{x}, \bar{b}, \bar{w} + 1/k) \in$ epi(F), which implies that $(\bar{b}, \bar{w} + 1/k)$ belongs to proj$_{(b,w)}$(epi(F)) and $(\bar{b}, \bar{w})$ belongs to cl(proj$_{(b,w)}$(epi(F))).

If F is convex, let $(\bar{b}, \bar{w})$ and $(\tilde{b}, \tilde{w})$ be two points in epi(f). Then, there exist sequences $\{\bar{x}_k\}$ and $\{\tilde{x}_k\}$ such that

$$F(\bar{x}_k, \bar{b}) \to f(\bar{b}) \leq \bar{w}, \qquad F(\tilde{x}_k, \tilde{b}) \to f(\tilde{b}) \leq \tilde{w}.$$

Using the definition of f and the convexity of F, we have for all $\alpha \in [0, 1]$ and k

$$\begin{aligned} f(\alpha\bar{b} + (1-\alpha)\tilde{b}) &\leq F(\alpha\bar{x}_k + (1-\alpha)\tilde{x}_k, \alpha\bar{b} + (1-\alpha)\tilde{b}) \\ &\leq \alpha F(\bar{x}_k, \bar{b}) + (1-\alpha)F(\tilde{x}_k, \tilde{b}). \end{aligned}$$

By taking the limit as $k \to \infty$, we obtain

$$f(\alpha\bar{b} + (1-\alpha)\tilde{b}) \leq \alpha f(\bar{b}) + (1-\alpha)f(\tilde{b}) \leq \alpha\bar{w} + (1-\alpha)\tilde{w}.$$

It follows that the point $\alpha(\bar{b}, \bar{w}) + (1-\alpha)(\tilde{b}, \tilde{w})$ belongs to epi(f). Thus epi(f) is convex, implying that f is convex. $\square$

We now provide a criteria for the *closedness* of partial minimization of a closed convex function F. If we guarantee that the projection of epi(F) is closed, then according to Theorem 3.10, the partial minimized function f is also closed. We resort to Theorem 2.6 about closedness of closed convex under linear transformation to prove that proj$_{(b,w)}$(epi(F)) is closed.

The following theorem is a version "with parameters" of Theorem 3.4 (page 44) on the existence of primal optimal solutions. Analogously, we still have an hypothesis of equality of lineality spaces and recession cones.

Theorem 3.11 (Optimal value of convex functions)**.** *Let $F : \mathbb{R}^{n+m} \to (-\infty, \infty]$ be a bounded below closed proper convex function, and let f be the function defined by*

$$f(b) = \inf_{x \in \mathbb{R}^n} F(x, b), \quad \text{for } b \in \mathbb{R}^m.$$

Assume that for some $\overline{b} \in \mathbb{R}^m$ and $\overline{\gamma} \in \mathbb{R}$ the set

$$L_{\overline{b},\overline{\gamma}} := \{x \in \mathbb{R}^n \mid F(x, \overline{b}) \leq \overline{\gamma}\}$$

is nonempty and its recession cone is equal to its lineality space. Then f is closed (proper, convex and bounded below). Furthermore, for each $b \in \operatorname{dom}(f)$, the set of minima in the definition of $f(b)$ is nonempty.

Proof. From Theorem 3.10, the function f is proper convex and bounded below, and the epigraph of f also satisfies the following relation:

$$\operatorname{proj}_{(b,w)}(\operatorname{epi}(F)) \subseteq \operatorname{epi}(f) \subseteq \operatorname{cl}(\operatorname{proj}_{(b,w)}(\operatorname{epi}(F)).$$

Thus, we only need to prove that the projection of $\operatorname{epi}(F)$ in the (b, w) variables is closed. Let us denote by A the projection on the variable (b, w). By Theorem 2.6, if the following condition is satisfied:

$$\ker(A) \cap \operatorname{recc}(\operatorname{epi}(F)) \subseteq \operatorname{lineal}(\operatorname{epi}(F)),$$

then the set $A \operatorname{epi}(F)$ is closed and its recession cone is equal to $A \operatorname{recc}(\operatorname{epi}(F))$. Note that A can be represented in the following matrix form:

$$A = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Therefore the kernel of A is equal to $\{(d_x, 0, 0) \in \mathbb{R}^{n+m+1} \mid d_x \in \mathbb{R}^n\}$, so the intersection $\ker(A) \cap \operatorname{recc}(\operatorname{epi}(F))$ is given by the recession directions of $\operatorname{epi}(F)$ with the form $(d_x, 0, 0)$. From the definition of the recession cone:

$$\operatorname{recc}(F) = \{(d_x, d_b) \in \mathbb{R}^{n+m} \mid (d_x, d_b, 0) \in \operatorname{recc}(\operatorname{epi}(F))\}.$$

Since the b-variable is fixed, the recession cone and lineality space of $F_{\bar{b}} := F(\bar{b}, \cdot)$ are given by

$$\begin{aligned}
\operatorname{recc}(F_{\bar{b}}) &= \{d_x \in \mathbb{R}^n \mid (d_x, 0, 0) \in \operatorname{recc}(\operatorname{epi}(F))\} \\
&= \ker(A) \cap \operatorname{recc}(\operatorname{epi}(F)), \\
\operatorname{lineal}(F_{\bar{b}}) &= \{d_x \in \mathbb{R}^n \mid (d_x, 0, 0) \in \operatorname{lineal}(\operatorname{epi}(F))\} \\
&= \ker(A) \cap \operatorname{lineal}(\operatorname{epi}(F)).
\end{aligned}$$

From theorem 3.2 (page 40), the recession cone and the lineality space of $L_{\bar{b},\bar{\gamma}}$ equals $\operatorname{recc}(F_{\bar{b}})$ and $\operatorname{lineal}(F_{\bar{b}})$, respectively, and from the hypothesis on $L_{\bar{b},\bar{\gamma}}$, $\operatorname{recc}(F_{\bar{b}})$ is equal to $\operatorname{lineal}(F_{\bar{b}})$. Then, we have

$$\ker(A) \cap \operatorname{recc}(\operatorname{epi}(F)) = \ker(A) \cap \operatorname{lineal}(\operatorname{epi}(F)) \subseteq \operatorname{lineal}(\operatorname{epi}(F)).$$

From Theorem 2.6 (page 21), we conclude that $A \operatorname{epi}(F)$ is closed with recession cone equal to $A \operatorname{recc}(\operatorname{epi}(F))$.

To prove the last claim, note that for all $b \in \operatorname{dom}(f)$ the problem

$$\begin{array}{ll} \min_x & F_b(x) \\ \text{s.t.} & x \in \mathbb{R}^n, \end{array}$$

has an optimal solution, since the level sets $L_{b,\gamma}$ have all the same recession cone, which implies that the recession cone of F_b is equal to its lineality space (Theorem 3.4, page 44). □

In the polyhedral case the result is trivial, since projection of a polyhedron is always a polyhedron. Note that the boundness assumption on F is important to ensure the existence of a x-optimal solution.

Corollary 3.12 (Optimal value of polyhedral functions)**.** *Let* $F : \mathbb{R}^{n+m} \to (-\infty, \infty]$ *be a bounded below polyhedral function, and let* f *be the function defined by*

$$f(b) = \inf_{x \in \mathbb{R}^n} F(x, b), \quad b \in \mathbb{R}^n.$$

Then f *is polyhedral and bounded below. Furthermore, for each* $b \in \mathrm{dom}(f)$*, the set of minima in the definition of* $f(b)$ *is nonempty.*

Proof. This Corollary follows from Theorem 3.10 (page 51) and the fact that projection of polyhedron is also polyhedron, see Corollary 2.9 (page 26). Moreover, the existence of optimal solution is guaranteed by Theorem 3.5. □

The closedness of a 0-1 MILP optimal value function f can be proved using the closedness of partial minimization of polyhedral functions. Actually, the epigraph of f is a finite union of polyhedrons.

Corollary 3.13 (Optimal value of mixed integer 0–1 programs)**.** *Let* A*,* G *be matrices and* c*,* q*,* b *be vectors, with appropriate dimensions. Assume that the optimal value of the following 0-1 mixed integer linear program*

$$\begin{array}{rl} f(b) = \min_{x,z} & c^\top x + q^\top z \\ s.t. & Ax + Gz \leq b, \\ & x \in \mathbb{R}^n, \ z \in \{0,1\}^l \end{array}$$

is bounded below, for all $b \in \mathrm{dom}(f)$*. Suppose also that* $\mathrm{dom}(f)$ *is nonempty. Then* f *piecewise polyhedral. Furthermore, for each* $b \in \mathrm{dom}(f)$*, the set of minima in the definition of* $f(b)$ *is nonempty.*

Proof. Let $z_i \in \{0,1\}^l$. The function f is the minimum of a finite number of polyhedral functions f_i:

$$\begin{array}{rl} f_i(b) = \min_{x} & c^\top x + q^\top z_i \\ \text{s.t.} & Ax \leq b - Gz_i, \\ & x \in \mathbb{R}^n. \end{array}$$

So, the epigraph of f is a *finite* union of polyhedral epigraph $\mathrm{epi}(f_i)$. $\square$

The following Corollary is a slight generalization of Corollary 3.13 for integer variables. However, we strongly believe that we do not need the assumption of finite number of feasible integer solution if we have that A and G are rational matrices. We did not find any reference for this result and even the original paper [Meyer] has slightly different hypothesis.

Corollary 3.14 (Optimal value of mixed integer programs)**.** *Let A, G be matrices and c, q, b be vectors with appropriate dimensions. Assume that the optimal value of the following mixed integer problem*

$$\begin{array}{rl} f(b) = \min_{x,z} & c^\top x + q^\top z \\ s.t. & Ax + Gz \leq b, \\ & x \in \mathbb{R}^n, \ z \in \mathbb{Z}^l \end{array}$$

is bounded below, for all $b \in \mathrm{dom}(f)$, and there is a finite number of feasible integer solutions z. Suppose also that $\mathrm{dom}(f)$ *is nonempty. Then f is piecewise polyhedral. Furthermore, for each $b \in \mathrm{dom}(f)$, the set of minima in the definition of $f(b)$ is nonempty.*

Proof. The proof is analogous to the proof of Corollary 3.13. $\square$

It is instructive to have an illustration of the fundamental difference between the properties of the optimal value functions from LPs and MILPs. Let A be a matrix, and c, b be vectors with appropriate dimension. Then we have two slightly distinct problems:

$$\begin{array}{rl} f(b) = \min & c^\top x \\ \text{s.t.} & a_i^\top x \leq b_i, \quad i = 1, \ldots, m, \\ & x \in X, \end{array}$$

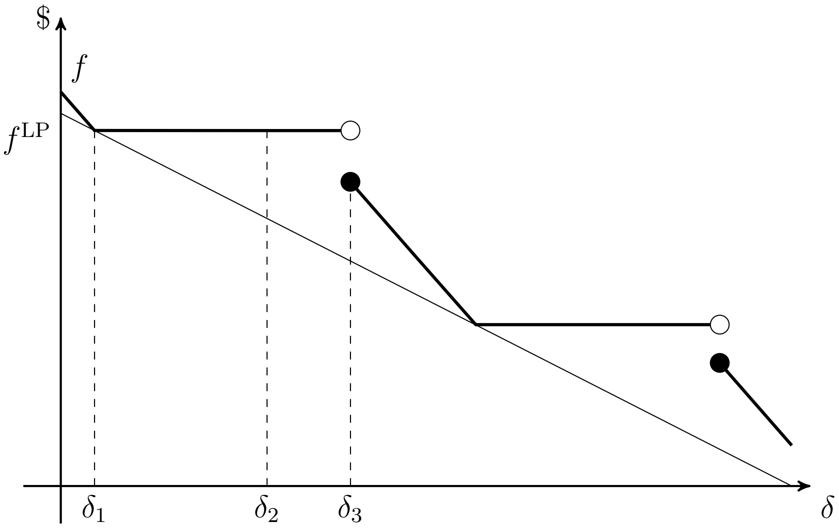
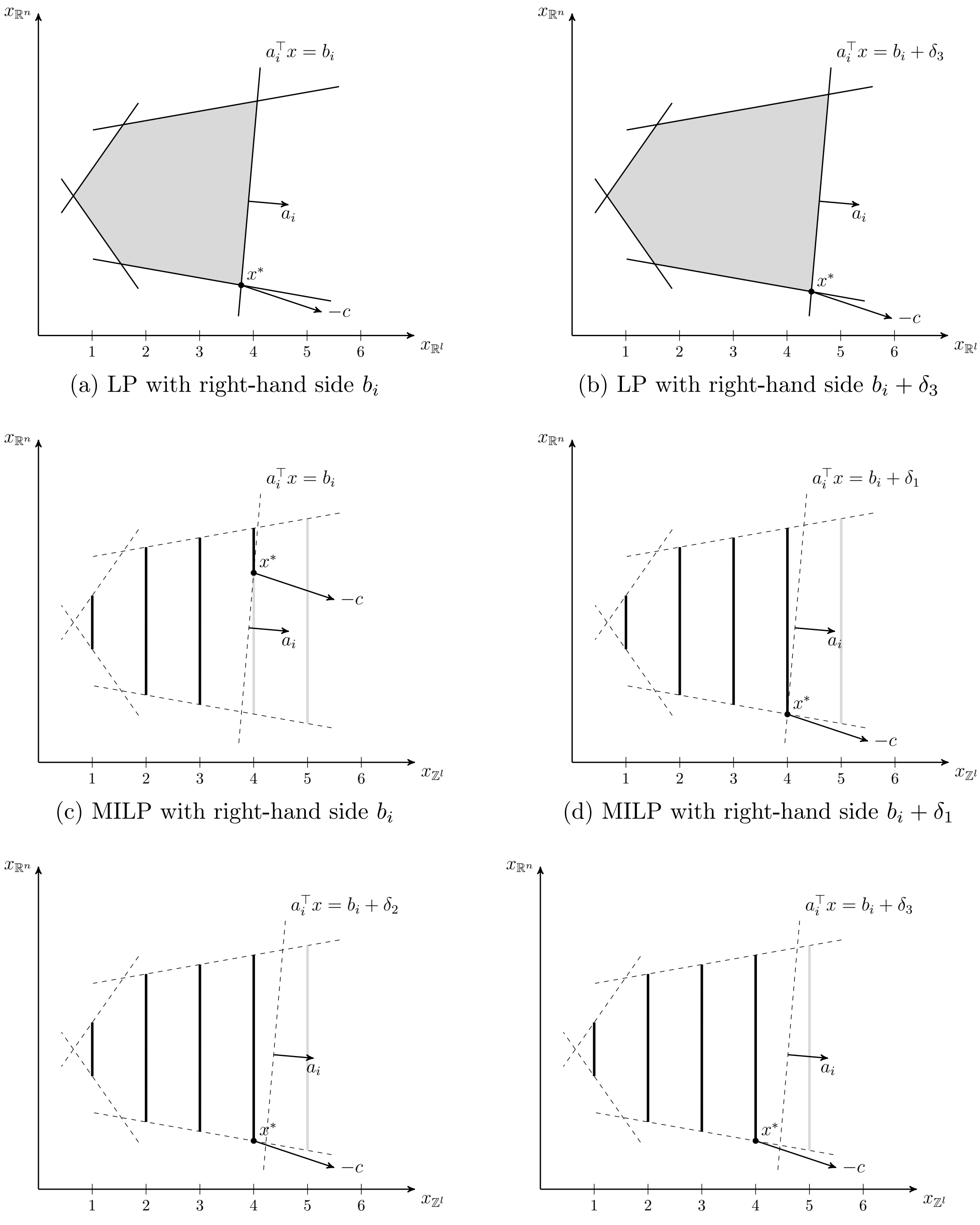


Figure 3.6: Sensitivity analysis of a MILP and LP problems

where $a_1, \ldots, a_m$ are the row vectors from A, and the set X is equal to $\mathbb{R}^{n+l}$ in the LP case or equal to $\mathbb{R}^n \times \mathbb{Z}^l$ in the MILP case. Note that each inequality constraint $a_i^\top x \leq b_i$ defines a half space, which is the opposite side of the hyperplane $a_i^\top x = b_i$ in the sense of the normal vector a_i, see figure 3.7c. Since the vector a_i specifies the direction and the scalar b_i specifies the position of the corresponding hyperplane, a small perturbation of the right-hand side b_i induces a small displacement of the hyperplane, but it does not change its direction (normal vector), see figure 3.7d. From the linear programming theory, there is an extreme point x^* of the feasible polyhedron which is the optimal solution of the LP, see figures 3.7c and 3.7d. Therefore, a small displacement of the hyperplanes that determine the extreme point x^* produces a small shift of x^*, which induces a small change in the optimal value. This illustrates the reason for the optimal value function $f(b)$ be continuous with respect to the right-hand side b in the LP case. In the MILP case, we have a different setting. If there is a finite number of integer feasible solutions, then the continuous feasible part belongs to a finite union of polyhedron, one polyhedron for each integer feasible solution. Let x^* be the optimal solution from the MILP problem, see figure 3.7e. For a small displacement of the hyperplanes, a new integer feasible solution may or may not be created. If there is no additional integer solution, then the analysis is analogous to the LP case. If a new integer solution is created, it could produce a sudden drop in the optimal value if the cost component in the integer direction is sufficiently smaller than the cost component in the continuous direction, see figure 3.7f. This illustrates why in the MILP case the optimal value function $f(b)$ is piecewise continuous with respect to the right-hand side b, with sudden drops at the discontinuity points.



(a) LP with right-hand side b_i

(b) LP with right-hand side $b_i + \delta_3$

(c) MILP with right-hand side b_i

(d) MILP with right-hand side $b_i + \delta_1$

(e) MILP with right-hand side $b_i + \delta_2$

(f) MILP with right-hand side $b_i + \delta_3$

Figure 3.7: Sensitivity analysis for LP and MILP.

3.4 Conjugacy, Duality and Lagrangian Relaxation

This section is about convex regularization of general functions and its relationship with Lagrangian Duality. We introduce a geometrical point of view to explain why support hyperplanes are the suitable objects to obtain convex hulls, the Conjugacy operation as a natural transformation to convexify a function, and an interpretation of Weak and Strong Duality in terms of the convex regularization of the primal optimal value function. In this last part, we also show the relationship between the biconjugate operation and the Lagrangian Relaxation of optimization problems.

Let D be a closed convex set. We note that D can be described as the intersection of all closed half spaces H that contains it, where the half space H is specified by a hyperplane tangent to D, called *support hyperplane*. Thus, the family of all support hyperplanes to D is sufficient to describe the points of D. We have two possibilities to parameterize support hyperplanes:

1. We can provide the boundary points x from D and then obtain the support hyperplane to D on x; or
2. We may specify a normal vector $\vec{v}$ and then obtain the support hyperplane with that given orthogonal vector to $\vec{v}$.

This second approach is the key to generate the convex regularization of a non-convex function.

Let f be a bounded below function, and assume that f is non-convex, i.e., the epigraph of f is a non-convex set. Let $y \in \mathbb{R}^n$ be a vector, and let r_{y,x_0} be the affine function with slope y that passes through the point $(x_0, f(x_0))$:

$$r_{y,x_0}(x) = y^\top (x - x_0) + f(x_0)$$

Note that the affine function r_{y,x_0} is equal to $f(x_0) - y^\top x_0$ at the origin. We may convince ourselves using figure 3.8 that the support hyperplane to epi(f) with slope y is the affine function r_y with the lowest intercept:

$$r_y(x) = y^\top x + \inf_{x\in\mathbb{R}^n} \{f(x) - y^\top x\}.$$

The intersection of all half-spaces containing the epigraph of f is equal to the

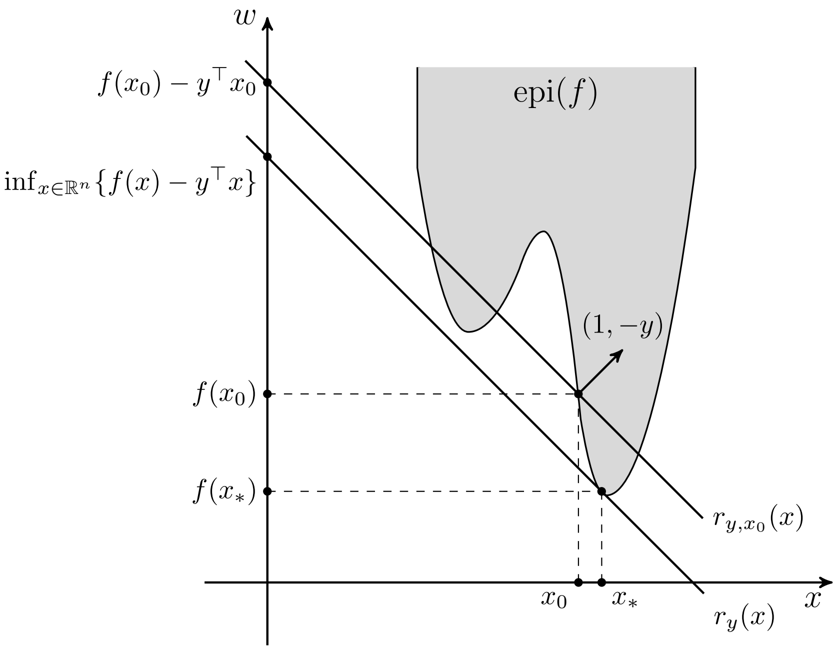


Figure 3.8: Support hyperplane to the epigraph of a function

closure of the convex hull of the corresponding epigraph. This resulting set defines a function $\check{f}$ which is the convex regularization of f, i.e., epi$(\check{f}) :=$ cl conv(epi(f)), see figure 3.9b. The hyperplane intersections that describes $\check{f}$ can be obtained by taking the pointwise supremum of all support hyperplanes r_y parametrized by y:

$$\check{f}(\overline{x}) = \sup_{y\in\mathbb{R}^n} r_y(\overline{x}), \qquad \forall x \in \mathbb{R}^n,$$

see figure 3.9a.

A related notion to the convex regularization is the conjugate function. It is defined, for all $y \in \mathbb{R}^n$, by

$$f^*(y) = \sup_{x \in \mathbb{R}^n} \{y^\top x - f(x)\}. \tag{3.3}$$

Note that $-f^*(y)$ is the intercept of the support hyperplane with slope y. Then, the convex regularization $\check{f}$ can be written as

$$\check{f}(x) = \sup_{y \in \mathbb{R}^n} \{y^\top x - f^*(y)\} = f^{**}(x),$$

in other words, the biconjugate of f is equal to the convex regularization $\check{f}$.

The function f^* is always a closed convex function, since its epigraph is the intersection of closed half spaces

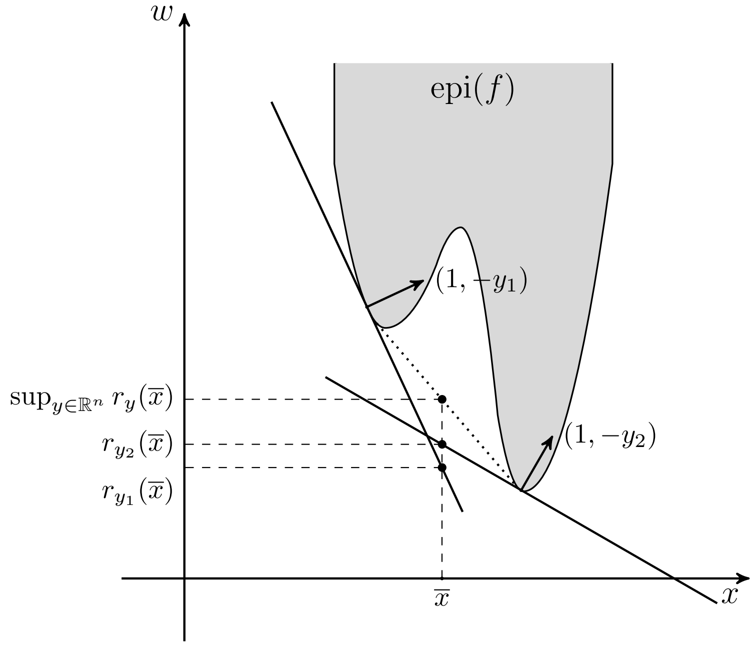
$$H_x = \{(y, w) \in \mathbb{R}^{n+1} \mid x^\top y - f(x) \leq w\}$$

parametrized by $x \in \mathbb{R}^n$. However, f^* can be proper or improper. Below, Theorem 3.15 states some other properties of the conjugate and biconjugate function. If f is proper and bounded below, then $\check{f}$ is also proper and bounded below. When $\check{f}$ is not proper, the functions f^{**} and $\check{f}$ may not coincide. For unbounded below functions, $\check{f}$ may be improper, even if f is proper. For instance, the function $f(x) = -|x|$ is proper, but the convex regularization is not: $\check{f} \equiv -\infty$.

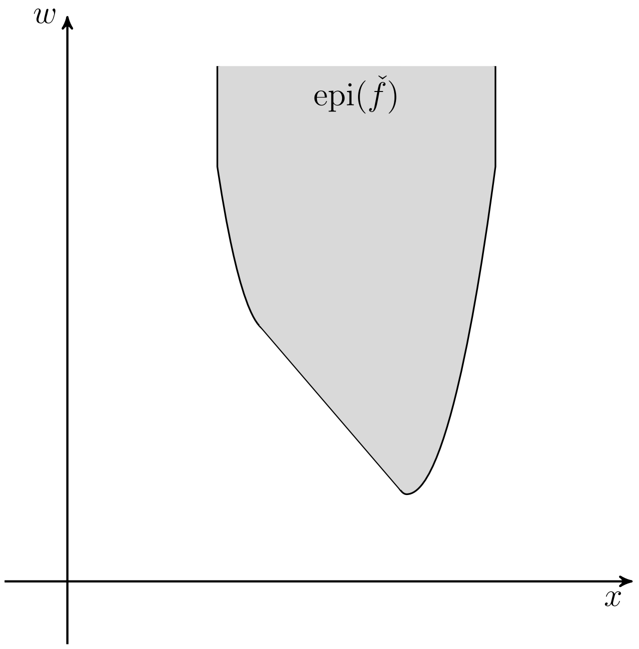
Theorem 3.15 (Conjugacy Theorem)**.** *Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a function, let f^* be its conjugate, and consider the biconjugate f^{**}. Then:*

1. *For all $x \in \mathbb{R}^n$,*

$$f(x) \geq f^{**}(x).$$



(a) Intersection of closed halfspaces contaning the epigraph.



(b) Convex regularization of a function.

Figure 3.9: Convex regularization of a nonconvex function.

2. *If f is convex, then properness of any one of the functions f, f^*, or f^{**} implies properness of the other two.*

3. *If f is closed, proper and convex, then*

$$f(x) = f^{**}(x), \qquad \forall x \in \mathbb{R}^n.$$

4. *If $\check{f}$ is proper, then*

$$\check{f}(x) = f^{**}(x), \qquad \forall x \in \mathbb{R}^n.$$

The proof of Theorem 3.15 is out of scope, see page 84 of [5]. Under the appropriate regularity conditions, we assume in the sequel that the convex regularization $\check{f}$ equals the biconjugate f^{**}, since it is more convenient to demonstrate the results of the following sections.

A traditional way of obtaining a convex regularization of a nonconvex programming problem is through the Lagrangian Relaxation. In fact, the biconjugate operation and the Lagrangian Relaxation of a mathematical programming problem have a close relationship which is established through the Theory of Lagrangian Duality.

We shall see below that the primal optimal value as a function of the right-hand side plays a central role in this link, since its biconjugate equals the Lagrangian Relaxation, which in turn is the dual optimal value of the corresponding dual problem. Through this relationship it is possible to interpret the weak/strong duality, and the existence of optimal dual solutions from a geometric point-of-view:

- the weak duality is a natural consequence of the convex regularization of the primal optimal value function;

- strong duality occurs when such regularization does not creates a gap with the primal optimal value function; and
- the optimal dual solution is a derivative of the convex regularization of the primal optimal value function.

These ideas are important to understand the Benders cut of SDDP and the Lagrangian cuts of SDDiP in a unified way.

Let $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^l$ be real-valued functions, and let D be a subset of $\mathbb{R}^n$. Consider the following optimization problem:

$$
\begin{aligned}
p(b, c) = \min_x \quad & f(x) \\
\text{s.t.} \quad & h(x) = b, \\
& g(x) \leq c, \\
& x \in D,
\end{aligned}
\tag{3.4}
$$

where $p(b, c)$ is the *primal optimal value* as a function of the right-hand sides (RHS) $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^l$. The *Lagrangian function* $L(x, \lambda, \mu)$ is defined by

$$
L(x, \lambda, \mu) = f(x) + \lambda^\top (h(x) - b) + \mu^\top (g(x) - c),
$$

the *dual function* is defined by

$$
\phi(\lambda, \mu) = \begin{cases} \inf_{x \in D} L(x, \lambda, \mu) & \text{if } \lambda \in \mathbb{R}^m \text{ and } \mu \in \mathbb{R}^l_+, \\ -\infty & \text{otherwise,} \end{cases}
$$

and the *dual problem* is given by

$$
\begin{aligned}
\max \quad & \phi(\lambda, \mu) \\
\text{s.t.} \quad & \lambda \in \mathbb{R}^m, \\
& \mu \in \mathbb{R}^l.
\end{aligned}
$$

The optimal value function of the dual problem, denoted by $w(b, c)$, is a function of the right-hand side parameters b and c. The dual problem is also called the *Lagrangian Relaxation*, when the primal one is a MILP. In this context, we refer to

$w(b, c)$ as the *Lagrangian Relaxation function*. The *weak duality* is the property of $w(b, c)$ being less than or equal to $p(b, c)$, and the *strong duality* is when both are equal. The weak duality holds for all RHS pair $(b, c) \in \mathbb{R}^{m+l}$, but strong duality usually requires additional regularity conditions (Slater condition, for instance).

We shall prove below the connection between the Lagrangian Relaxation and the biconjugate of the primal optimal value function $p(b, c)$. For this purpose, we have to represent the dual function $\phi(\lambda, \mu)$ in the following form:

$$\begin{array}{rl} \phi(\lambda, \mu) = \inf_{x,\bar{b},\bar{c}} & \{f(x) + \lambda^\top(\bar{b} - b) + \mu^\top(\bar{c} - c)\} \\ \text{s.t.} & h(x) = \bar{b} \\ & g(x) \leq \bar{c} \\ & x \in D. \end{array} \tag{3.5}$$

Indeed, if μ has a negative component, then $\phi(\lambda, \mu)$ is equal to $-\infty$, and so is the right-hand side of the equation (3.5), since we can increase arbitrarily the corresponding component of $\bar{c}$. If all components of μ are non-negative, then increasing $\bar{c}$ also increases the objective function $f(x) + \lambda^\top(\bar{b} - b) + \mu^\top(\bar{c} - c)$. So, the optimal solution for $\bar{c}$ is $g(x)$. Since $\bar{b}$ is equal to $h(x)$, the right-hand side of (3.5) equals $\inf_{x \in D} L(x, \lambda, \mu)$. Therefore, dual function $\phi(\lambda, \mu)$ can be represented in the form (3.5).

By changing the minimization order of (3.5), we can restate the dual function $\phi(\lambda, \mu)$ in terms of the conjugate of $p(b, c)$:

$$\begin{aligned} \phi(\lambda, \mu) &= \inf_{\bar{b},\bar{c}} \left\{ \inf_{\substack{x \in D, \\ h(x)=\bar{b}, \\ g(x) \leq \bar{c}.}} \{f(x)\} + \lambda^\top(\bar{b} - b) + \mu^\top(\bar{c} - c) \right\} \\ &= \inf_{\bar{b},\bar{c}} \{p(\bar{b}, \bar{c}) + \lambda^\top \bar{b} + \mu^\top \bar{c}\} - \lambda^\top b - \mu^\top c \\ &= -p^*(-\lambda, -\mu) - \lambda^\top b - \mu^\top c. \end{aligned}$$

The dual optimal solution $w(b, c)$ is the supremum of $\phi(\lambda, \mu)$ on (λ, μ), which is the biconjugate of $p(b, c)$:

$$\begin{aligned}\sup_{\lambda,\mu} \phi(\lambda, \mu) &= \sup_{\lambda,\mu}\{-p^*(-\lambda, -\mu) + (-\lambda, -\mu)^\top (b, c)\}\\ &= \sup_{\lambda,\mu}\{(\lambda, \mu)^\top (b, c) - p^*(\lambda, \mu)\} = p^{**}(b, c),\end{aligned}$$

where the second equality is obtained by changing $(-\lambda, -\mu)$ for (λ, μ), respectively. Therefore, Weak Duality occurs because the biconjugate $p^{**}(b, c)$ of the primal optimal value function $p(b, c)$ is always a lower bound, for every pair (b, c), and Strong Duality occurs when there is no gap between the convex regularization $p^{**}(b, c)$ and the primal optimal function $p(b, c)$. We note that the dual optimal solutions $(\overline{\lambda}, \overline{\mu})$ are the maximizers in the definition of the biconjugate $p^{**}(b, c)$:

$$(\overline{\lambda}, \overline{\mu}) \in \arg\max_{\lambda,\mu} \phi(\lambda, \mu) \quad\Longleftrightarrow\quad (-\overline{\lambda}, -\overline{\mu}) \in \arg\max_{\lambda,\mu}\{(\lambda, \mu)^\top (b, c) - p^*(\lambda, \mu)\}. \tag{3.6}$$

This is an important remark for computing "derivatives" in section 3.5.

The following results characterize the biconjugate of the primal optimal value function for convex, polyhedral and MILP problems. We recall that when the primal optimal value function equals the corresponding biconjugate, then we have strong duality.

Corollary 3.16 (Biconjugate of convex optimal value function)**.** *Let $F : \mathbb{R}^{n+m} \to (-\infty, \infty]$ be a bounded below closed proper convex function, and let f be the function defined by partial minimization*

$$f(b) = \inf_{x\in\mathbb{R}^n} F(x, b), \quad b \in \mathbb{R}^n.$$

Assume that for some $\overline{b} \in \mathbb{R}^n$ and $\overline{\gamma} \in \mathbb{R}$ the set

$$L_{\overline{b},\overline{\gamma}} = \{x \in \mathbb{R}^m \mid F(x, \overline{b}) \le \overline{\gamma}\}$$

is nonempty and its recession cone is equal to its lineality space. Then the biconjugate function f^{**} is equal to f.

Proof. From Theorem 3.11, we have that f is a closed proper convex function, and from Theorem 3.15 item (c) we have that the biconjugate of a closed proper convex function is equal to the original function. $\square$

Corollary 3.17 (Biconjugate of polyhedral optimal value function)**.** *Let $F : \mathbb{R}^{n+m} \to (-\infty, \infty]$ be a bounded below polyhedral function, and let f be the function defined by partial minimization*

$$f(b) = \inf_{x \in \mathbb{R}^n} F(b, x), \quad b \in \mathbb{R}^n.$$

Then the biconjugate function f^{**} is equal to f.

Proof. From Corollary 3.12 we have that f is a polyhedral function, and from Theorem 3.15 item (c) we conclude that f^{**} is equal to f. $\square$

The following characterization of the biconjugate of a MILP problem is an important an important result for the Blessing of Binary from the SDDiP, see section 6.2.

Corollary 3.18 (Optimal value of mixed integer programs)**.** *Let A, G be rational matrices, and c, q, b be vectors with appropriate dimensions, and $X \subset \mathbb{R}^n \times \mathbb{Z}^l$ be a set such that $\operatorname{conv}(X)$ is a polyhedron. Assume that the optimal value of the following mixed integer program*

$$\begin{array}{rl} f(b) = \min_{x,z} & c^\top x + q^\top z \\ s.t. & Ax + Gz \leq b, \\ & (x, z) \in X, \end{array}$$

is bounded below, for all $b \in \operatorname{dom}(f)$. Suppose also that $\operatorname{dom}(f)$ is nonempty. Then the biconjugate of f is equal to

$$\begin{array}{rl} f^{**}(b) = \min_{x,z} & c^\top x + q^\top z \\ s.t. & Ax + Gz \leq b, \\ & (x, z) \in \operatorname{conv}(X). \end{array}$$

Proof. Denote by $\overline{f}(b)$ the optimal value function from the problem with constraint $\mathrm{conv}(X)$. Let $\phi(\mu)$ and $\overline{\phi}(\mu)$ be the dual functions from the corresponding problems with constraint X and $\mathrm{conv}(X)$, respectively:

$$\phi(\mu) = \inf_{(x,z)\in X} \{c^\top x + q^\top z + \mu^\top (Ax + Gz - b)\},$$
$$\overline{\phi}(\mu) = \inf_{(x,z)\in \mathrm{conv}(X)} \{c^\top x + q^\top z + \mu^\top (Ax + Gz - b)\},$$

for all $\mu \geq 0$. By Lemma 3.7, we have that $\phi = \overline{\phi}$, and since the biconjugate of an optimal value function is equal to the corresponding dual optimal value we have that

$$f^{**}(b) = \sup_{\mu\geq 0} \phi(\mu) = \sup_{\mu \geq 0} \overline{\phi}(\mu) = \overline{f}^{**}(b).$$

Since $\overline{f}$ is polyhedral, we conclude that

$$\overline{f}^{**} = \overline{f}. \qquad \square$$

3.5 Subgradient, chain rule and optimality condition

In this section, we present a more general notion of derivative, the *subgradient*, that is crucial for algorithms that solve non-differentiable convex programming problems. We also show how to compute subgradients of optimal value functions, the chain rule for some convex preserving compositions and some optimality conditions.

Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a proper convex function. We say that a vector $g \in \mathbb{R}^n$ is a *subgradient* of f at a point $x \in \mathrm{dom}(f)$ if

$$f(y) \geq f(x) + g^\top (y - x), \quad \forall y \in \mathbb{R}^n.$$

The set of all subgradients of f at x is called the *subdifferential* of f at x and is denoted by $\partial f(x)$. By convention, $\partial f(x)$ is considered empty for all $x \notin \mathrm{dom}(f)$.

As figure 3.10 illustrates, the vector g is a subgradient of f at x if and only if the hyperplane in $\mathbb{R}^{n+1}$ that has normal $v = (g, -1)$ and passes through $(x, f(x))$ supports the epigraph of f. Indeed, by definition, the vector $g \in \mathbb{R}^n$ is a subgradient of f at x if and only if

$$\begin{aligned} v^\top(x, f(x)) &\geq v^\top(y, f(y)), \\ &\geq v^\top(y, w), \quad \forall (y, w) \in \operatorname{epi}(f), \end{aligned}$$

which is equivalent to

$$v^\top(y, w) \leq b, \quad \forall (y, w) \in \operatorname{epi}(f),$$

where b is defined as $v^\top(x, f(x))$, see figure 3.10. If f is differentiable at x, then the only subgradient is $\nabla f(x)$. For instance, let f be the modulus function. Note

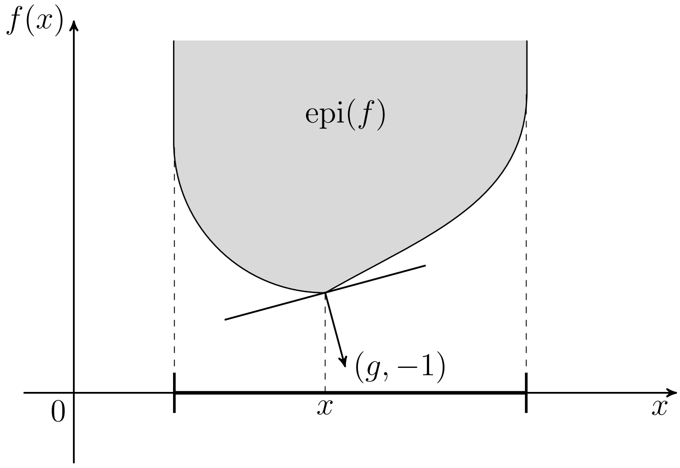


Figure 3.10: Subgradient of a convex function.

that $f(x) = -x$ for all x less than 0, and $f(x) = x$ for all x greater than 0. For those points, f is differentiable and $\nabla f(x)$ is equal to -1 and 1, respectively, see figure 3.11. For x equal to 0, the function f is not differentiable (left and right derivatives are different), but there are subgradients. From the definition, the subgradients at 0 are those $g \in \mathbb{R}$ such that

$$|y| \geq g^\top y, \quad \forall y \in \mathbb{R}. \tag{3.7}$$

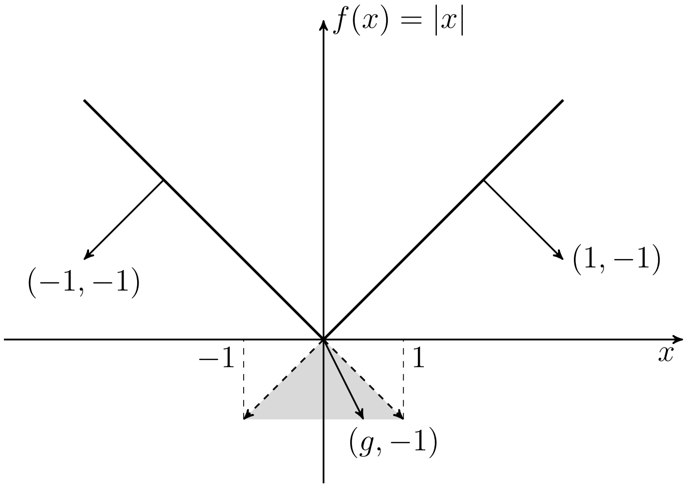


Figure 3.11: Subgradient of the modulus function.

Note that the only scalars g that satisfy to (3.7) for all $y \in \mathbb{R}$ are those from $[0, 1]$. Therefore, the subdifferential of f at 0 is $\partial f(0) = [-1, 1]$, see figure 3.11.

In practice, the computation of subdifferential of a given convex function f is not an easy task, specially using just the subgradient definition. We present below the Conjugate Subgradient Theorem that provides us an attractive way to compute subgradients for the optimal value function of convex programs.

Theorem 3.19 (Conjugate Subgradient Theorem)**.** *Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a proper convex function and let f^* be its conjugate. The following three relations are equivalent for a pair of vectors $(\overline{x}, \overline{g})$:*

(i) $f(\overline{x}) + f^*(\overline{g}) = \overline{g}^\top \overline{x}$*;*

(ii) $\overline{x} \in \arg\max_x \{\overline{g}^\top x - f(x)\}$*;*

(iii) $\overline{g} \in \partial f(\overline{x})$*.*

If in addition f is closed, the relations (i), (ii) and (iii) are equivalent to *(iv)* $\overline{g} \in \arg\max_g\{g^\top \overline{x} - f^*(g)\}$;

(v) $\overline{x} \in \partial f^*(\overline{g})$.

Proof. $(i) \Rightarrow (ii)$: If a pair $(\overline{x}, \overline{g})$ satisfies equation (i), then $\overline{x}$ attains the supremum in the definition $f^*(\overline{g}) = \sup_x\{\overline{g}^\top x - f(x)\}$.
$(ii) \Rightarrow (iii)$: If $\overline{x}$ is the maximizer from the conjugate definition, then the following inequality holds:

$$\overline{g}^\top \overline{x} - f(\overline{x}) \geq \overline{g}^\top x - f(x), \quad \forall x \in \mathbb{R}^n, \tag{3.8}$$

which is equivalent to $\overline{g}$ be a subgradient of f at $\overline{x}$.
$(iii) \Rightarrow (i)$: If $\overline{g}$ is a subgradient of f at $\overline{x}$, we have

$$\overline{g}^\top \overline{x} - f(\overline{x}) \geq \sup_{x \in \mathbb{R}^n}\{\overline{g}^\top x - f(x)\} = f^*(\overline{g}),$$

so equation (i) holds.

By the Conjugacy Theorem (Theorem 3.15-3), if f is closed then the biconjugate f^{**} is equal to f. Let us denote by h the conjugate function f^*. Thus, h^* is equal to f, and the relation (i) is equivalent to

$$h^*(\overline{x}) + h(\overline{g}) = \overline{x}^\top \overline{g}.$$

From relations (ii) and (iii), we conclude that (i) is equivalent to $\overline{g} \in \arg\max_g\{\overline{x}^\top g - h(g)\}$ and $\overline{x} \in \partial h(\overline{g})$. Replacing the notation h by the definition f^*, we conclude that (i) is equivalent to (iv) and (v). $\square$

The following Corollary provides an interesting formula to computing subgradients for optimal value functions of convex programs. We recall that $\overline{g}$ is an optimal solution of $\max_g\{g^\top \overline{x} - f^*(g)\}$ if and only if $-\overline{g}$ is an optimal solution of the corresponding dual optimal problem, see equation (3.6) from page 67.

Corollary 3.20 (Subdifferential of convex function)**.** *Let* $f : \mathbb{R}^n \to (-\infty, \infty]$ *be a proper closed convex function. Then,*

$$\overline{g} \in \partial f(\overline{x}) \iff \overline{g} \in \arg\max_g \{g^\top \overline{x} - f^*(g)\}. \tag{3.9}$$

Proof. This is the equivalence between items (*iii*) and (*iv*) from Theorem 3.19. □

We can extend the previous Corollary for Lagrangian Relaxation functions, that is, for the optimal value function of the Lagrangian dual of a MILP. This will be useful for solving multistage MILP problems.

Corollary 3.21 (Subdifferential of convex regularization)**.** *Let* $f : \mathbb{R}^n \to (-\infty, \infty]$ *be a bounded below proper function. Then,*

$$\overline{g} \in \partial f^{**}(\overline{x}) \iff \overline{g} \in \arg\max_g \{g^\top \overline{x} - f^*(g)\}. \tag{3.10}$$

Proof. From Theorem 3.19 - items (iii) and (iv), we have that

$$\overline{g} \in \partial f^{**}(\overline{x}) \iff \overline{g} \in \arg\max_{g \in \mathbb{R}^n} \{g^\top \overline{x} - (f^{**})^*(g)\}.$$

Since f^* is convex, the properness of f^{**} implies the properness of f^*, by Theorem 3.15-(2). By Theorem 3.15-(3), f^{***} is equal to f^*, so we conclude the result. □

Until now, we have seen the relationship between subgradients of optimal value functions and dual optimal solutions. However, there are optimization problems with equal primal and dual optimal values, but no dual optimal solution. For an example, take

$$\begin{array}{rl} \min & x \\ \text{s.t.} & x^2 \leq 0. \end{array}$$

Then $x^* = 0$ is the only feasible/optimal solution, and we have

$$q(\mu) = \inf_{x \in \mathbb{R}} \{x + \mu x^2\} = -\frac{1}{4\mu}, \qquad \forall \mu > 0,$$

and $q(\mu) = -\infty$ for $\mu \leq 0$, so that $q^* = f^* = 0$. However, there is no $\mu^* \geq 0$ such that $q(\mu^*) = q^* = 0$. This is a typical constrained optimization situation where there are no dual optimal solutions.

The following Theorem presents some sufficient conditions for the existence of dual optimal solutions, however its proof requires Farkas Lemma and the Nonlinear Farkas Lemma, which is out of the scope of this document. For a proof, see pages 169 and 173 of Bertsekas.

Theorem 3.22 (Existence of dual optimal solution)**.** *Consider the following problem*

$$\begin{array}{rl} \min_x & f(x) \\ s.t. & h(x) = 0 \\ & g(x) \leq 0, \\ & x \in X, \end{array}$$

where $X \subset \mathbb{R}^n$ is a convex set, $h : X \to \mathbb{R}^m$ is an affine function, $g : X \to \mathbb{R}^l$ and $f : X \to \mathbb{R}$ are convex functions. Assume that f^* is finite and that one of the following three conditions holds:

1. *There exists $\overline{x} \in X$ such that $h(\overline{x}) = 0$ and $g(\overline{x}) < 0$, and there exists $\tilde{x} \in \mathrm{ri}(X)$ such that $h(\tilde{x}) = 0$;*
2. *The function g is affine, and there exists $\overline{x} \in \mathrm{ri}(X)$ such that $h(\overline{x}) = 0$ and $g(\overline{x}) \leq 0$;*
3. *The function f and g are affine, and X is a polyhedron.*

Then the primal and dual optimal values are equal, and the set of optimal solutions of the dual problem is nonempty.

The interior condition (1) in the preceding proposition is known in the nonlinear programming literature as the Slater condition.

The following lemma establishes a relationship between primal and dual optimal solutions that is useful for the proof of the chain rule for subdifferentials. Note that equation (3.11) remind the definition of a Lagrangian Multiplier μ^* associated to stationary point x^*. Actually, it is an extension of the Lagrangian Multiplier definition for non-differentiable convex programs. In this context, we say that (x^*, μ^*) is a primal-dual optimal solution pair if x^* is the optimal solution of a given primal problem and μ^* is the optimal solution of the corresponding dual problem.

Lemma 3.23 (Dual solution and Lagrangian Multiplier)**.** *Consider the optimization problem*

$$\begin{array}{rl} \min_x & f(x) \\ s.t. & g(x) \leq 0, \\ & x \in X, \end{array}$$

where $X \subset \mathbb{R}^n$ is a convex set, $g : X \to \mathbb{R}^m$ and $f : X \to \mathbb{R}$ are convex functions. Strong duality holds and (x^*, μ^*) is primal-dual optimal solution pair if and only if x^* is feasible, $\mu^* \geq 0$, and

$$x^* \in \arg\min_{x \in X} L(x, \mu^*), \quad \mu^{*\top} g(x^*) = 0. \tag{3.11}$$

Proof. Let us denote by f^* and q^* the primal and dual optimal values, respectively. If $f^* = q^*$, and x^* and μ^* are primal and dual optimal solutions, respectively, then we have the following relations:

$$\begin{aligned} f^* = q^* &= \sup_{\mu \in \mathbb{R}^m} \phi(\mu) = \phi(\mu^*) \\ &= \inf_{x \in X} L(x, \mu^*) \leq L(x^*, \mu^*) \\ &= f(x^*) + \sum_{i=1}^{m} \mu_i^* g_i(x^*) \leq f(x^*) = f^*, \end{aligned}$$

where the last inequality follows from the feasibility of x^*, $g(x^*) \leq 0$, and the non-negativity of μ^*, i.e., $\mu^* \geq 0$. Therefore, $L(x^*, \mu^*) = \inf_{x \in X} L(x, \mu^*)$ and $\mu_i^* g_i(x^*) = 0$, for all $i = 1, \ldots, m$.

Conversely, suppose that x^* is feasible, $\mu^* \geq 0$, and equation (3.11) is satisfied. Then, we have

$$q^* \geq q(\mu^*) = f(x^*) + \sum_{i=1}^{m} \mu_i^* g_i(x^*) = f(x^*) \geq f^* \geq q^*.$$

Therefore, the primal and dual optimal values are equal, $f^* = q^*$, and (x^*, μ^*) is a primal and dual optimal solution pair. $\square$

The concept of subgradient is defined for convex functions, and so the results for the chain rule are restricted to operations that preserve convexity. For a long list of convexity-preserving rules, see section 3.2 of Boyd. Here, we focus on two convexity-preserving operations which are important for stochastic programming: pre-composition with linear mappings, and sums of convex functions.

Theorem 3.24 (Pre-composition with linear mapping)**.** *Let $f : \mathbb{R}^m \to (-\infty, \infty]$ be a convex function, let A be an $m \times n$ matrix, and assume that the function F given by*

$$F(x) = f(Ax)$$

is proper, that is, assume that f is proper and $\operatorname{Im} A \cap \operatorname{dom}(f) \neq \emptyset$*. Then*

$$\partial F(x) \supseteq A^\top \partial f(x), \qquad \forall x \in \mathbb{R}^n.$$

Furthermore, if either f is polyhedral or the range of A contains a point in the relative interior of $\operatorname{dom}(f)$*, i.e.,* $\operatorname{Im} A \cap \operatorname{ri}(\operatorname{dom}(f)) \neq \emptyset$*, we have*

$$\partial F(x) = A^\top f(Ax), \qquad \forall x \in \mathbb{R}^n.$$

Proof. Note that $\operatorname{dom}(F) = A^{-1}(\operatorname{dom}(f))$. If $x \notin \operatorname{dom}(F)$, then both $\partial F(x) = \emptyset$ and $A^\top \partial f(Ax) = \emptyset$. Let $\overline{x} \in \operatorname{dom}(F)$. If $d \in A^\top \partial f(A\overline{x})$, there exists $g \in \partial f(A\overline{x})$

such that $d = A^\top g$. We have for all $x \in \mathbb{R}^n$,

$$\begin{aligned} F(x) - F(\overline{x}) - d^\top(x - \overline{x}) &= f(Ax) - f(A\overline{x}) - (A^\top g)^\top (x - \overline{x}) \\ &= f(Ax) - f(A\overline{x}) - g^\top (Ax - A\overline{x}) \\ &\geq 0, \end{aligned}$$

where the last inequality follows from g being a subgradient of f. Therefore, $A^\top \partial f(A\overline{x}) \subset \partial F(\overline{x})$.

To prove the reverse inclusion under the given assumption, we let $d \in \partial F(\overline{x})$ and we show that $d \in A^\top \partial f(A\overline{x})$ by viewing x as the solution of an optimization problem induced by the subgradient definition. For all $x \in \mathbb{R}^n$

$$F(x) \geq F(\overline{x}) + d^\top (x - \overline{x}),$$

or equivalently

$$f(Ax) - d^\top x \geq f(A\overline{x}) - d^\top \overline{x}.$$

Therefore, the pair $(\overline{x}, A\overline{x})$ is an optimal solution of the following optimization problem

$$\begin{array}{rl} \min_{x,y} & f(y) - d^\top x \\ \text{s.t.} & Ax = y, \\ & (x, y) \in \mathbb{R}^n \times \operatorname{dom}(f). \end{array} \tag{3.12}$$

If f is a polyhedral function, (3.12) can be reformulated as a linear programming problem. If the intersection $\operatorname{Im} A \cap \operatorname{ri}(\operatorname{dom}(f))$ is nonempty, then there is a pair $(\tilde{x}, \tilde{y})$ that belongs to $\mathbb{R}^n \times \operatorname{ri}(\operatorname{dom}(f))$ such that $A\tilde{x} = \tilde{y}$. In either case, we conclude from Theorem 3.22 (item 2, page 74) that there is no duality gap, and that there is a dual optimal solution $\overline{\lambda} \in \mathbb{R}^m$ associated to the equality constraint $Ax = y$. By Theorem 3.23, the vector $(\overline{x}, A\overline{x})$ is the optimal solution of the Lagrangian problem:

$$\min_{(x,y) \in \mathbb{R}^n \times \operatorname{dom}(f)} \{f(y) - d^\top x + \overline{\lambda}^\top (Ax - y)\}.$$

Since the minimization over x is unconstrained, we must have $d = A^\top \overline{\lambda}$. Thus, $A\overline{x}$ is an optimal solution of

$$\min_{y \in \operatorname{dom}(f)} \{f(y) - \lambda^\top y\},$$

or equivalently

$$f(y) \geq f(A\overline{x}) + \lambda^\top (y - A\overline{x}), \quad \forall y \in \mathbb{R}^m.$$

Hence $\overline{\lambda} \in \partial f(A\overline{x})$, so $d = A^\top \overline{\lambda} \in A^\top \partial f(A\overline{x})$. Therefore, $\partial F(\overline{x}) = A^\top \partial f(A\overline{x})$. $\square$

The conditions of Corollary 3.25 can be further extended to consider the sum of a mixture of polyhedral and convex functions. However, the purpose of this document is to emphasize the additional requirements for proving a given result in the convex case in comparison to the polyhedral case, not to be an extensive reference on the subject. For a more general condition, see page 194 from Bertsekas.

Corollary 3.25 (Sum of convex functions)**.** *Let* $f_i : \mathbb{R}^n \to (-\infty, \infty]$, $i = 1, \ldots, m$, *be convex functions, and assume that the function* $F = f_1 + \cdots + f_m$ *is proper, that is,* $\bigcap_{i=1}^m \mathrm{dom}(f_i) \neq \emptyset$*. Then*

$$\partial F(x) \supseteq \partial f_1(x) + \cdots + \partial f_m(x), \qquad \forall x \in \mathbb{R}^n.$$

Furthermore, if all functions f_i *are polyhedral or* $\bigcap_{i=1}^m \mathrm{ri}(\mathrm{dom}(f_i)) \neq \emptyset$*, we have equality.*

Proof. Let $y = (y_1, \ldots, y_m) \in \mathbb{R}^{n \cdot m}$ and f defined by $f(y) = f_1(y_1) + \cdots + f_m(y_m)$. Then, one can easily check from the definition that

$$\partial f(y) = \partial f_1(y_1) \times \cdots \times \partial f_m(y_m).$$

Let A be the linear map that creates m copies of a vector $x \in \mathbb{R}^n$:

$$A = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix}, \quad Ax = (x, \ldots, x)^\top.$$

Note that $F(x) = f(Ax)$. If each f_i is polyhedral, f is also polyhedral. If the intersection $\cap_{i=1}^m \mathrm{ri}(\mathrm{dom}(f_i))$ is nonempty, $\mathrm{Im}\, A \cap \mathrm{ri}(\mathrm{dom}(f))$ is nonempty. Indeed, let $\tilde{x} \in \cap_{i=1}^{m} \operatorname{ri}(\operatorname{dom}(f_i))$ and let $\tilde{y} = (\tilde{x}, \dots, \tilde{x})$. Hence, $\tilde{y}$ belongs to $\operatorname{Im} A$ and

$$\begin{aligned}\operatorname{ri}(\operatorname{dom}(f)) &= \operatorname{ri}\Big(\operatorname{dom}(f_1) \times \cdots \times \operatorname{dom}(f_m)\Big)\\ &= \operatorname{ri}(\operatorname{dom}(f_1)) \times \cdots \times \operatorname{ri}(\operatorname{dom}(f_m)),\end{aligned}$$

so $\tilde{y}$ also belongs to $\operatorname{ri}(\operatorname{dom}(f))$. Therefore, $\operatorname{Im} A \cap \operatorname{ri}(\operatorname{dom}(f))$ is nonempty. In both cases, by the conditions of Theorem 3.24, we get

$$\begin{aligned}F(x) &= A^\top \partial f(x)\\ &= \begin{bmatrix} I & \cdots & I \end{bmatrix} \Big(\partial f_1(x) \times \cdots \times \partial f_m(x)\Big)\\ &= \partial f_1(x) + \cdots + \partial f_m(x).\end{aligned}$$

□

We now present an optimality condition for non-smooth convex programming problems. This is a natural extension of the differentiable case, where we have a stationary point $x^* \in X$ if and only if $-\nabla f(x^*)$ belongs to the normal cone of X at x^*, i.e., if and only if any direction that locally decrease the objective function f points out of the feasible region X, see page 31 and 73 of Solodov 1. The normal cone of X at x^* is given by the directions d that have an angle wider than $90°$ with all feasible directions from X at x^*:

$$N_X(x^*) = \{d \in \mathbb{R}^n \mid (x - x^*)^\top d \leq 0,\ \forall x \in X\}.$$

We can easily verify from the subgradient definition that the subdifferential of an indicator function $\delta_X(x)$ is the normal cone $N_X(x)$. The following Corollary proves that a negative subgradient $-g \in -\partial f(x^*)$ belongs to the normal cone $N_X(x^*)$ if and only if the point $x^* \in X$ is an optimal solution for the problem of minimizing f over X.

Corollary 3.26 (Optimality condition)**.** *Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a proper convex function, let X be a nonempty convex subset of $\mathbb{R}^n$. Consider the optimization problem*

$$\begin{array}{rl}\min & f(x)\\ s.t. & x \in X,\end{array}$$

and assume that one of the following two conditions holds:

1. *f and X are polyhedral, and $\mathrm{dom}(f) \cap X \neq \emptyset$; or*
2. *$\mathrm{ri}(\mathrm{dom}(f)) \cap \mathrm{ri}(X) \neq \emptyset$.*

Then, a vector x^* minimizes f over X if and only if

$$-\partial f(x^*) \cap N_X(x^*) \neq \emptyset,$$

where $N_X(x^*)$ is the normal cone at $x^* \in X$.

Proof. Let $F = f + \delta_X$, where δ_X is the indicator function of X. If f is a polyhedral function and X is a polyhedral set, then F is a polyhedral function. If $\mathrm{ri}(\mathrm{dom}(f)) \cap \mathrm{ri}(X) \neq \emptyset$, we have that

$$\mathrm{ri}(\mathrm{dom}(f)) \cap \mathrm{ri}(\mathrm{dom}(\delta_X)) = \mathrm{ri}(\mathrm{dom}(f)) \cap \mathrm{ri}(X) \neq \emptyset.$$

In both cases, we are in the conditions of Corollary 3.25. Thus,

$$\partial(f + \delta_X)(x) = \partial f(x) + \partial \delta_X(x).$$

Note that if $g \in \partial \delta_X(\overline{x})$, then $\overline{x} \in X$ and also

$$0 \geq g^\top (x - \overline{x}), \quad \forall x \in X,$$

since $\delta_X(x)$ is equal to 0 if $x \in X$, and equal to ∞ otherwise. Hence, $\partial_X(x) = N_X(x)$. Additionally, a vector x^* minimizes f over X if and only if 0 is a subgradient of $\partial(f + \delta_X)$:

$$f(x) + \delta_X(x) \geq f(x^*) + \delta_X(x^*) + 0^\top (x - x^*), \quad \forall x \in \mathbb{R}^n.$$

We conclude that x^* minimizes f over X if and only if the vector 0 belongs to $\partial f(x^*) + N_X(x^*)$. $\square$

4 DECISION UNDER UNCERTAINTY

The purpose of this chapter is to review some concepts of stochastic optimization: dynamic programming, scenario tree and the SDDP algorithm, to guide the presentation of the SDDiP algorithm in Chapter 6.

4.1 Stochastic dynamic programming

Consider a planning problem in which $T \in \mathbb{Z}_+$ decisions must be taken sequentially along time. Denote by $x_t \in \mathbb{R}^{n_t}$ the vector that represents the decision at time t, also called stage t. Suppose that the associated constraints are linear,

$$Ax_1 = b_1, \ x_1 \geq 0,$$

as well as the influence of the decision of the previous stage in the decision of the current stage:

$$B_t x_{t-1} + A_t x_t = b_t, \ x_t \geq 0.$$

Suppose also that the total decision cost is a linear function:

$$c_1^\top x_1 + \cdots + c_T^\top x_T.$$

If all the parameters of this model are known, the planning problem is *deterministic* and is described by a simple linear program (LP):

$$\begin{array}{llllllll}
\min\limits_{x_1,x_2,\ldots,x_T} & c_1^\top x_1 & + & c_2^\top x_2 & + & \cdots & + & c_T^\top x_T \\
\text{s.t.} & A_1 x_1 & & & & & & = b_1, \\
 & B_2 x_1 & + & A_2 x_2 & & & & = b_2, \\
 & & & & \ddots & & & \vdots \\
 & & & & & B_T x_{T-1} + & A_T x_T & = b_T, \\
 & x_1 \geq 0, & & x_2 \geq 0, & & \cdots & x_T \geq 0. &
\end{array} \tag{4.1}$$

A relevant class of stochastic optimization problems with the deterministic structure (4.1) is obtained by considering the right-hand sides $b_t \in \mathbb{R}^{m_t}$ as a stochastic process. In this case, each decision x_t should be adapted to the random process realization:

$$\text{decision}(x_1) \rightsquigarrow \text{observation}(b_2) \rightsquigarrow \text{decision}(x_2) \rightsquigarrow \\ \ldots \rightsquigarrow \text{observation}(b_T) \rightsquigarrow \text{decision}(x_T).$$

More formally, each decision x_t is a function of all the past observations up to time t, so we denote $x_t = x_t(b_{[t]})$, where $b_{[t]} = (b_1, b_2, \ldots, b_t)$. Note that we assume b_1 to be known, so the first decision is deterministic.

Since they depend on the random process $\{b_t\}_{t=1}^T$, each function $x_t(b_{[t]})$ is referred to as a *decision rule* or *policy*.

As the total cost is also a random variable, it is necessary to take a statistic of it, for example the expected value, to obtain a criterion for choosing a decision rules. So, we set our stochastic optimization problem as:

$$\begin{array}{lll} \min\limits_{x_1, x_2(\cdot), \ldots, x_T(\cdot)} & \mathbb{E}\left[c_1^\top x_1 + c_2^\top x_2(b_{[2]}) + \cdots + c_T^\top x_T(b_{[T]})\right] & \\ \text{s.t.} & B_t x_{t-1}(b_{[t-1]}) + A_t x_t(b_{[t]}) = b_t, \quad t = 1, \ldots, T, & (4.2) \\ & \qquad\qquad\qquad x_t(b_{[t]}) \geq 0, \quad t = 1, \ldots, T, & \end{array}$$

where we omit the term $B_1 x_0$ at the first stage. However, this formulation is not suitable for large-scale problems since it requires evaluating each possible realization of the process $\{b_t\}_{t=1}^T$, which implied a very high computational complexity to obtain the optimal policies. This motivates the introduction of the dynamic programming formulation, which is suitable for computing “good” feasible policies.

We now proceed with the derivation of the dynamic programming formulation following the ideas of SHAPIRO [25]. One delicate argument of this derivation is the interchangeability between the expectation and minimum operators. So, we first consider the case of a discrete random variable b, where we denote by p_i the probability of the outcome $b = b^i$. In this way, the interchangeability principle is given by the following identity between optimal values of different optimization problems:

$$\begin{array}{rl} \min\limits_{x^1,\dots,x^N} & \sum\limits_{i=1}^{N} p_i \cdot c^\top x^i \\ \text{s.t.} & Ax^i = b^i, \quad i = 1,\dots,N, \\ & x^i \geq 0, \qquad i = 1,\dots,N, \end{array} = \sum_{i=1}^{N} p_i \cdot \left[\begin{array}{rl} \min\limits_{x^i} & c^\top x^i \\ \text{s.t.} & Ax^i = b^i, \\ & x^i \geq 0, \end{array} \right] \tag{4.3}$$

Identity (4.3) occurs because the objective function is separable and there is no constraint coupling the decisions $x^1, \dots, x^N$. The interchangeability principle also holds in a more general case, replacing the weighted average by the expected value:

$$\begin{array}{rl} \min\limits_{x(\cdot)} & \mathbb{E}\left[c^\top x\right] \\ \text{s.t.} & Ax(b) = b, \\ & x(b) \geq 0, \end{array} = \mathbb{E}\left[\min_{\substack{Ax=b,\\ x\geq 0}} c^\top x\right]. \tag{4.4}$$

We note that in the left-hand side of equation (4.4) the decision variable x is a random vector (depending on b) and in the right-hand side the decision variable x is a vector of $\mathbb{R}^n$, but we now have independent minimization problems for each realization of b. Intuitively, the interchangeability principle states that it is equivalent to obtain the optimal random variable directly or to construct the optimal random variable from every possible optimal outcome.

Thus, consider the stochastic programming problem (4.2). Using the Law of Total Expectation, we can represent (4.2) in the following form:

$$\begin{array}{rl} \min\limits_{x_1, x_2(\cdot), \dots, x_T(\cdot)} & \mathbb{E}_{|b_{[1]}}\left[\mathbb{E}_{|b_{[2]}}\left[\cdots \mathbb{E}_{|b_{[T-1]}}\left[c_1^\top x_1 + c_2^\top x_2(b_{[2]}) + \cdots + c_T^\top x_T(b_{[T]})\right]\right]\right] \\ \text{s.t.} & B_t x_{t-1}(b_{[t-1]}) + A_t x_t(b_{[t]}) = b_t, \quad t = 1,\dots,T, \\ & x_t(b_{[t]}) \geq 0, \quad t = 1,\dots,T, \end{array} \tag{4.5}$$

where $\mathbb{E}_{|b_{[t-1]}}$ is the conditional expectation with respect to the probability of $b_{[t]}$ given the realization $b_{[t-1]}$ of the random process up to time $t-1$. By the interchangeability principle, the formulation (4.5) is equivalent to

$$\min_{\substack{A_1x_1=b_1\\x_1\geq 0}} c_1^\top x_1 + \mathbb{E}_{|b_1}\left[\min_{\substack{B_2x_1+A_2x_2=b_2\\x_2\geq 0}} c_2^\top x_2 + \mathbb{E}_{|b_{[2]}}\left[\cdots + \mathbb{E}_{|b_{[T-1]}}\left[\min_{\substack{B_Tx_{T-1}+A_Tx_T=b_T\\x_T\geq 0}} c_T^\top x_T\right]\right]\right] \tag{4.6}$$

where we also used that $x_t(b_{[t]})$ is constant given $b_{[t]}$ so we can pull it out from the conditional expectations. We call equation (4.6) the *nested formulation.* Note that in each stage t, the minimization problem inside the expectation $\mathbb{E}_{|b_{[t-1]}}$ has the decision x_{t-1} and the right-hand side b_t fixed, as well as all previous history $b_{[t-1]}$ so we don't need to decide on a function anymore.

The *dynamic programming formulation* is an equivalent way of presenting stochastic optimization problems which is convenient for describing some algorithms and examples. The idea of dynamic programming is to build a recursive relationship between optimal value functions, where the function at the end of the recursion usually considers the whole problem. In order to create this recursion, we use the nested formulation (4.6) to define the *future cost-to-go* function $\overline{\mathcal{Q}}_t$ from stage t:

$$\overline{\mathcal{Q}}_t(x_{t-1}, b_{[t-1]}) = \mathbb{E}_{|b_{[t-1]}}\left[\min_{\substack{B_tx_{t-1}+A_tx_t=b_t\\x_t\geq 0}} c_t^\top x_t + \cdots + \mathbb{E}_{|b_{[T-1]}}\left[\min_{\substack{B_Tx_{T-1}+A_Tx_T=b_T\\x_T\geq 0}} c_T^\top x_T\right]\right], \tag{4.7}$$

for $t = 2, \ldots, T$. So the future cost-to-go function $\overline{\mathcal{Q}}_t(x_{t-1}, b_{[t-1]})$ is the expected cost from stage t onwards, given the decision x_{t-1} and conditioned on the realization $b_{[t-1]}$. Note that $\overline{\mathcal{Q}}_t(\cdot,\cdot)$ satisfies the following recurrence relation:

$$\overline{\mathcal{Q}}_t(x_{t-1}, b_{[t-1]}) = \mathbb{E}_{|b_{[t-1]}}\left[\min_{\substack{B_tx_{t-1}+A_tx_t=b_t\\x_t\geq 0}} c_t^\top x_t + \overline{\mathcal{Q}}_{t+1}(x_t, b_{[t]})\right], \tag{4.8}$$

for $t = 2, \ldots, T$, where we define $\overline{\mathcal{Q}}_{T+1}(\cdot,\cdot)$ as the identically zero function. Thus, the problem for the first decision x_1 can be described as

$$\min_{\substack{A_1x_1=b_1\\x_1\geq 0}} c_1^\top x_1 + \overline{\mathcal{Q}}_2(x_1, b_1). \tag{4.9}$$

To complete the usual dynamic programming definition, it is necessary to define the *cost-to-go* function $Q_t(\cdot,\cdot)$ of stage t:

$$Q_t(x_{t-1}, b_{[t]}) := \min_{\substack{B_tx_{t-1}+A_tx_t=b_t\\x_t\geq 0}} c_t^\top x_t + \overline{\mathcal{Q}}_{t+1}(x_t, b_{[t]}), \quad t = 1,\ldots,T. \tag{4.10}$$

The cost-to-go function $Q_t(\cdot,\cdot)$ can be interpreted as the lowest cost obtained by a compromise between the immediate cost $c_t^\top x_t$ and the future cost-to-go function $\overline{\mathcal{Q}}_{t+1}(x_t, b_{[t]})$. Note that the cost-to-go function $Q_t(x_{t-1}, b_{[t]})$ also appears inside the expectation on (4.8):

$$\overline{\mathcal{Q}}_t(x_{t-1}, b_{[t-1]}) = \mathbb{E}_{|b_{[t-1]}}\left[Q_t(x_{t-1}, b_{[t]})\right], \quad t = 2,\ldots,T, \tag{4.11}$$

which means that the future cost-to-go function in stage t is the expected value of the cost-to-go function of time t. Thus, we can state the *stochastic dynamic programming* formulation as:

$$\begin{aligned} Q_t(x_{t-1}, b_{[t]}) &= \min_{\substack{B_tx_{t-1}+A_tx_t=b_t\\x_t\geq 0}} c_t^\top x_t + \overline{\mathcal{Q}}_{t+1}(x_t, b_{[t]}), \quad t \in \{1,\ldots,T\} \\ \overline{\mathcal{Q}}_{t+1}(x_t, b_{[t]}) &= \begin{cases} \mathbb{E}_{|b_{[t]}}\left[Q_{t+1}(x_t, b_{[t+1]})\right] &, \quad t \in \{1,\ldots,T-1\} \\ 0 &, \quad t = T \end{cases}. \end{aligned} \tag{4.12}$$

Note that the last function of the recursion is the cost-to-go function $Q_1(b_1)$, which is the optimal value of (4.9). If we expand the definition of each cost-to-go and future cost-to-go functions we recover the nested formulation (4.6).

In this dissertation, we focus on the particular case where $\{b_t\}_{t=1}^T$ is a stage-wise independent process (SWI), that is, we assume that b_t is independent of $b_{[t-1]}$, for every stage t. In particular, we have that the conditional expectation $\mathbb{E}_{|b_{[t]}}[\cdot]$

equals the unconditional one, i.e., $\mathbb{E}_{|b_{[t]}}[\cdot] = \mathbb{E}[\cdot]$. Since the last cost-to-go function is given by

$$Q_T(x_{T-1}, b_T) = \min_{\substack{B_T x_{T-1} + A_T x_T = b_T \\ x_T \geq 0}} c_T^\top x_T,$$

and by the stagewise independence property, the future cost-to-go function of stage T satisfies the equation $\overline{\mathcal{Q}}_T(x_{T-1}, b_{[T-1]}) = \mathbb{E}_{b_{[T-1]}}[Q_T(x_{T-1}, b_T)] = \mathbb{E}[Q_T(x_{T-1}, b_T)]$. So, it only depends on x_{T-1}, that is, $\overline{\mathcal{Q}}_T(x_{T-1}, b_{[T-1]}) \equiv \overline{\mathcal{Q}}_T(x_{T-1})$. Repeating this argument backwards in time, suppose that the cost-to-go function of stage t is given by

$$Q_t(x_{t-1}, b_t) = \min_{\substack{B_t x_{t-1} + A_t x_t = b_t \\ x_t \geq 0}} c_t^\top x_t + \overline{Q}_{t+1}(x_{t-1}).$$

Then, we have a similar identity for the future cost-to-go function: $\overline{\mathcal{Q}}_t(x_{t-1}, b_{[t-1]}) = \mathbb{E}_{b_{[t-1]}}[Q_t(x_{t-1}, b_t)] = \mathbb{E}[Q_t(x_{t-1}, b_t)]$. So, we just have a single future cost-to-go function $\overline{\mathcal{Q}}(\cdot)$ per stage, that is, the future cost-to-go function $\overline{\mathcal{Q}}$ does not depend on the realization of $b_{[t]}$. We can state the dynamic formulation for the stagewise independent case as:

$$Q_t(x_{t-1}, b_t) = \min_{\substack{B_t x_{t-1} + A_t x_t = b_t \\ x_t \geq 0}} c_t^\top x_t + \overline{\mathcal{Q}}_{t+1}(x_t), \quad t \in \{1, \ldots, T\} \tag{4.13}$$

$$\overline{\mathcal{Q}}_{t+1}(x_t) = \begin{cases} \mathbb{E}[Q_{t+1}(x_t, b_{t+1})] & , \quad t \in \{1, \ldots, T-1\} \\ 0 & , \quad t = T \end{cases} .$$

We will see in the next sections that the SDDP algorithm iteratively computes approximations for $\overline{\mathcal{Q}}_{t+1}(x_t)$ and these approximations induce a feasible policy.

4.2 Scenario tree for stagewise independent process (SWI)

In order to solve (4.12), it is necessary to evaluate in certain points the future cost-to-go function $\overline{\mathcal{Q}}_{t+1}(x_t)$ and the corresponding derivatives (subgradients). However, the distribution of b_t is, in general, an absolutely continuous distribution, which means that calculating the expectation is equivalent to solving an integral that has no obvious antiderivative. An alternative to this problem is to approximate the distribution of the random process by a finite distribution called a *scenario tree*. There are several techniques to perform this approximation: Monte Carlo, quasi-Monte Carlo, importance sampling, Latin Hypercube, among others. See SHAPIRO; DENTCHEVA; RUSZCZYŃSKI [26] and RUSZCZYNSKI [23] for a reference in the stochastic programming context.

The purpose of this section is to present the notation for a stagewise independent scenario tree (SWI), comment on the corresponding symmetries of the cost-to-go and future cost-to-go functions, and analyze the properties of these functions based on the results of chapter 3. This section is important to understand the algorithm presented in the next one.

Let $\{b_t\}_{t=1}^T$ be a SWI discrete stochastic process, that is, suppose that the random variable b_t can take any value in $b_t^1, \dots, b_t^{N_t}$ with probability $p_t^1, \dots, p_t^{N_t}$, respectively, and that b_t does not depend on past observations $b_{[t-1]}$, i.e.,

$$\mathbb{P}[b_t = b_t^i \mid b_{[t-1]}] = \mathbb{P}[b_t = b_t^i], \qquad \forall i = 1, \dots, N_t,$$

for all $t = 1, \dots, T$. So, the future cost-to-go function $\overline{\mathcal{Q}}_t(x_{t-1})$ is a weighted average of the cost-to-go functions $Q_t(x_{t-1}, b_t)$:

$$\overline{\mathcal{Q}}_t(x_{t-1}) = \mathbb{E}\left[Q_t(x_{t-1}, b_t)\right] = \sum_{i=1}^{N_t} p_t^i \cdot Q_t(x_{t-1}, b_t^i). \tag{4.14}$$

Thus, the dynamic programming formulation of the SWI case is given by:

$$Q_t(x_{t-1}, b_t^i) = \min_{\substack{B_t x_{t-1} + A_t x_t = b_t^i \\ x_t \geq 0}} c_t^\top x_t + \overline{\mathcal{Q}}_{t+1}(x_t) \tag{4.15}$$

$$\overline{\mathcal{Q}}_{t+1}(x_t) = \begin{cases} \sum_{i=1}^{N_t} p_t^i \cdot Q_{t+1}(x_t, b_t^i) & , t \in \{1, \dots, T-1\} \\ 0 & , t = T \end{cases}$$

for all $i = 1, \dots, N_t$ and $t = 1, \dots, T$. Note that in the last stage, the cost-to-go function $Q_T(x_{T-1}, b_T)$ is an optimal value function of a LP, since the future cost-to-go function $\overline{\mathcal{Q}}_{T+1}$ is identically zero. From Corollary 3.12 (page 55), the cost-to-go function of the last stage $Q_T(\cdot, b_T^i)$ is polyhedral, as well as the future cost-to-go function of the last stage, because $\overline{\mathcal{Q}}_T(x_{T-1})$ is a convex combination of polyhedral functions. Since the optimal value function of a polyhedral function is polyhedral, again by Corollary 3.12, and convex combinations of polyhedral functions are also polyhedral, by backward induction on stage all the cost-to-go and future cost-to-go functions are polyhedral. Moreover, by Corollary 3.20 (page 73) about the subdifferential of optimal value functions and by Theorem 3.24 (page 76) about chain rule for subdifferentials, a vector g is a subgradient of the cost-to-go function $Q_t(x_{t-1}, b_t^i)$ with respect to x_{t-1} if it is of the form $g = B^\top \pi$, where π is a dual optimal solution of (4.15) associated to the constraint $B_t x_{t-1} + A_{t-1} = b_t^i$. By Corollary 3.25, we conclude that all subgradients g of the future cost-to-go function $\overline{\mathcal{Q}}_{t+1}(x_t)$ are of the form $g = \sum_{i=1}^{N_t} p_t^i g_t^i$, where g_t^i are subgradients of $Q_{t+1}(x_t, b_t^i)$ with respect to x_t.

However, the subgradient of any of those functions depends on the cost-to-go and future cost-to-go functions of the next stage, but we do not know it *a priori*. Only in the last stage it is possible to compute subgradients using those formulas, since the future cost-to-go function of stage $T + 1$ is identically zero. We shall see in the next section how the SDDP algorithm obtains cuts for these functions.

A final comment concerns the symmetry of the SWI scenario tree and how this is reflected for both future cost-to-go and cost-to-go functions. The Figure 4.1 presents a SWI scenario tree, and figures 4.2a and 4.2b show the cost-to-go and future cost-to-go from each node, respectively. Note that there is a single future cost-to-go function per stage and as many cost-to-go functions as the branching of the tree node. If the scenario tree were dependent, there would be different future cost-to-go and cost-to-go functions per node. Therefore, a SWI process simplifies the complexity of the stochastic optimization problem and is therefore a common requirement for large scale multistage stochastic problem, so it is computationally feasible to solve them in practice. In the next section, we shall see how the SDDP algorithm approximates the future cost-to-go function $\overline{\mathcal{Q}}_t(\cdot)$ iteratively.

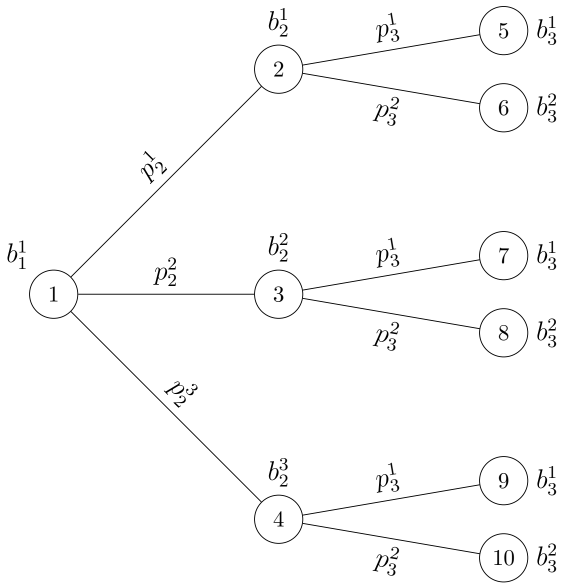
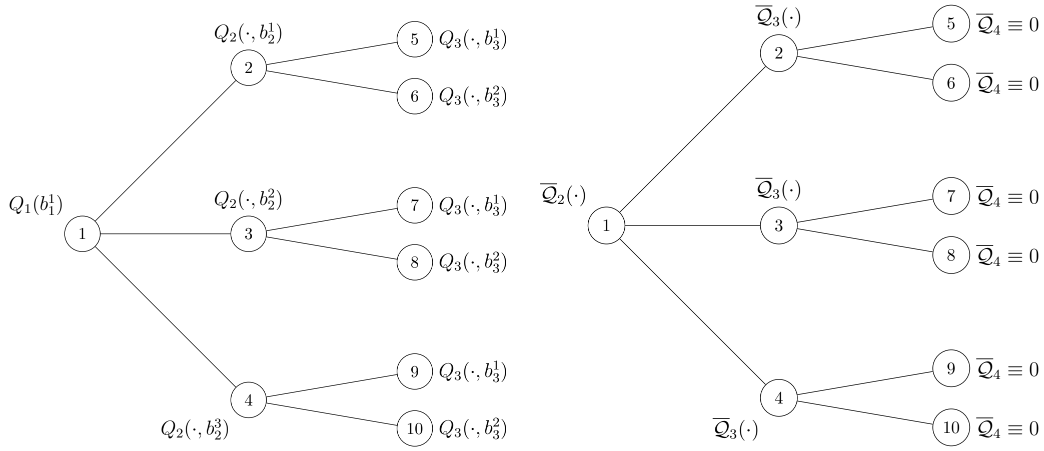


Figure 4.1: Stagewise independent scenario tree.



(a) Cost-to-go functions. (b) Future cost-to-go functions.

Figure 4.2: Cost-to-go and future cost-to-go functions for a SWI problem.

4.3 The Stochastic Dual Dynamic Programming (SDDP) algorithm

The idea of the SDDP algorithm is to create a family of polyhedral functions $\overline{\mathfrak{Q}}_t(\cdot)$ that approximate from below the future cost-to-go functions $\overline{\mathcal{Q}}_t(\cdot)$, that is,

$$\overline{\mathfrak{Q}}_t(x_{t-1}) \leq \overline{\mathcal{Q}}_t(x_{t-1}), \quad \forall x_{t-1} \in \mathbb{R}^{n_{t-1}},$$

for all $t = 2, \ldots, T$. The SDDP refines this approximation in such a way that in the next iteration the new polyhedral function $\overline{\mathfrak{Q}}_t(\cdot)$ is greater than or equal to the previous one $\overline{\mathfrak{Q}}_t^{\text{old}}(\cdot)$, but always below the future cost-to-go function $\overline{\mathcal{Q}}_t$, see figure 4.3. This monotonicity requirement of the lower approximation is crucial to guarantee the convergence of the method. From the future cost-to-go approximation $\overline{\mathfrak{Q}}_{t+1}(\cdot)$, we can also obtain an approximation of the cost-to-go function of the previous stage

$$\mathfrak{Q}_t(x_{t-1}, b_t^i) := \min_{\substack{B_t x_{t-1} + A_t x_t = b_t^i \\ x_t \geq 0}} c_t^\top x_t + \overline{\mathfrak{Q}}_{t+1}(x_t), \quad \forall x_{t-1}. \tag{4.16}$$

The approximation (4.16) is important to find points $x_t \in \mathbb{R}^{N_t}$ around which the cuts for $\overline{\mathfrak{Q}}_{t+1}(x_t)$ will the calculated. We note that the function $\mathfrak{Q}_t(x_{t-1}, b_t^i)$ is less

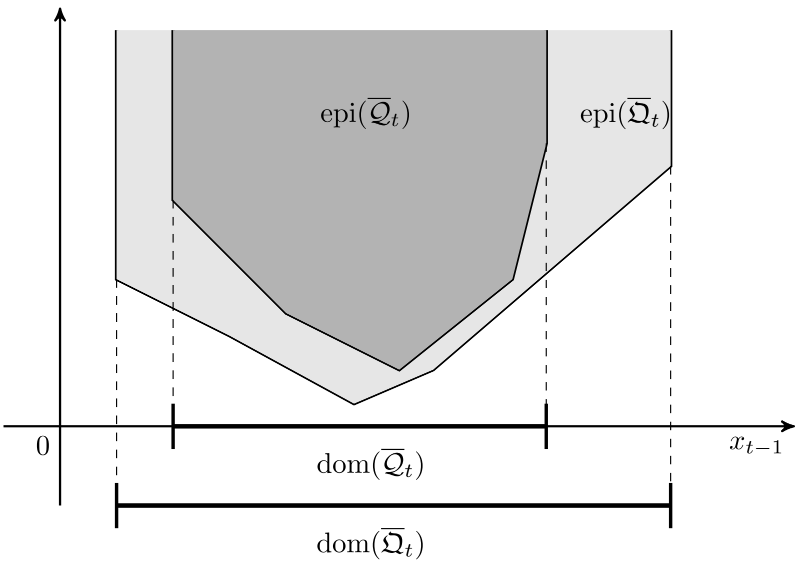


Figure 4.3: Future cost-to-go function approximation.

than or equal to $Q_t(x_{t-1}, b_t^i)$, since the corresponding optimization problems have the same constraints, but the objective function from the former is smaller than or equal to the latter.

We assume that the future cost-to-go functions $\overline{\mathcal{Q}}_t(\cdot)$ are bounded below, i.e., there is $L \in \mathbb{R}$ such that $\overline{\mathcal{Q}}_t(x_{t-1}) \geq L$, for all $x_{t-1} \in \mathbb{R}^{N_{t-1}}$ and every stage $t = 2, \ldots, T$. Thus, the SDDP algorithm starts with the approximation $\overline{\mathfrak{Q}}_t(\cdot)$ who is equal to the constant L.

To refine the approximation of the future cost-to-go function $\overline{\mathfrak{Q}}_t(\cdot)$ around a point x_{t-1}^*, one simply computes subgradients g_t^i for $\mathfrak{Q}_t(\cdot, b_t^i)$ at x_{t-1}^*, and note that the following relations hold for all $x_{t-1} \in \mathbb{R}^{n_{t-1}}$:

$$\begin{aligned}
\overline{\mathcal{Q}}_t(x_{t-1}) &= \sum_{i=1}^{N_t} p_t^i \cdot Q_t(x_{t-1}, b_t^i) \\
&\geq \sum_{i=1}^{N_t} p_t^i \cdot \mathfrak{Q}_t(x_{t-1}, b_t^i) \\
&\geq \sum_{i=1}^{N_t} p_t^i \cdot \left(\mathfrak{Q}_t(x_{t-1}^*, b_t^i) + g_t^{i\top}(x_{t-1} - x_{t-1}^*)\right) \\
&= \sum_{i=1}^{N_t} p_t^i \cdot \mathfrak{Q}_t(x_{t-1}^*, b_t^i) + \left[\sum_{i=1}^{N_t} p_t^i g^i\right]^\top \cdot (x_{t-1} - x_{t-1}^*), \qquad (4.17) \\
&\forall x_{t-1} \in \mathbb{R}^{N_{t-1}},
\end{aligned}$$

where the first identity follows from (4.14), the second relation follows from $\mathfrak{Q}_t(\cdot,\cdot)$ being less than or equal to $Q_t(\cdot,\cdot)$, the third relation follows from the definition of subgradient, and the fourth just reorganizes the terms of the affine function. Thus, the new approximation of $\overline{\mathfrak{Q}}_t(\cdot)$ is the maximum between the old future cost-to-go function $\overline{\mathfrak{Q}}_t^{\text{old}}(\cdot)$ and the affine function (4.17):

$$\overline{\mathfrak{Q}}_t(x_{t-1}) := \max\left\{\overline{\mathfrak{Q}}_t^{\text{old}}(x_{t-1}), \sum_{i=1}^{N_t} p_t^i \cdot \mathfrak{Q}_t(x_{t-1}^*, b_t^i) + \left[\sum_{i=1}^{N_t} p_t^i g_t^i\right]^\top \cdot (x_{t-1} - x_{t-1}^*)\right\}.$$

It is instructive to note that the maximum of polyhedral functions is again a polyhedral function, and in the last stage, $t = T$, the function (4.17) represents a support hyperplane for the future cost-to-go function $\overline{\mathcal{Q}}_T(\cdot)$, since the lower approximation $\mathfrak{Q}_T(\cdot,\cdot)$ equals the cost-to-go function $Q_T(\cdot,\cdot)$. This refinement is known as the Backward step of the SDDP, since it improves $\overline{\mathfrak{Q}}_t$ from $\mathfrak{Q}_t$, which depends on $\overline{\mathfrak{Q}}_{t+1}$.

Finally, it is necessary to define a way to choose the points x_t^* around which a linear approximation will be obtained for the future cost-to-go function $\overline{\mathcal{Q}}_{t+1}(\cdot)$, as suggested in (4.17). One approach is to sample a path in the SWI scenario tree to obtain realizations $b_1^*, \ldots, b_T^*$ for the right-hand sides of each stage and then compute the optimal solution $x_t^* \in \mathbb{R}^{n_t}$ of (4.16) in each stage t:

$$x_t^* \in \underset{\substack{B_t x_{t-1}^* + A_t x_t = b_t^* \\ x_t \geq 0}}{\arg\min} \; c_t^\top x_t + \overline{\mathfrak{Q}}_{t+1}(x_t, b_t^i),$$

where the previous stage solution x_{t-1}^* is used in x_{t-1}. This is known as the Forward step of SDDP, since we choose decisions x_t going forward on the tree. The need to sample paths from the scenario tree instead of going through all the possibilities lies in the fact that the number of paths increases exponentially with the size T of the planning horizon.

In short, the SDDP algorithm has two steps and a stopping criterion:

i) **Forward step**: find feasible solutions x_t^* for $t = 1, \ldots, N_t$.

ii) **Backward step**: compute lower linear approximations of the future cost-to-go functions centered on the feasible solutions x_{t-1}^* obtained in the forward step, in decreasing stage order. After refining all approximations of the cost-to-go and future cost-to-go functions, we estimate a lower bound $\underline{z} := \mathfrak{Q}_1(b^1)$ for the first stage.

iii) **Stopping criterion**: If either a maximum number of iterations is reached, or the optimal value $\underline{z}$ makes small relative progress, stop. Otherwise, do another Forward/Backward step.

The proof of convergence of the SDDP uses that the cost-to-go and future cost-to-go functions are polyhedral functions, see PHILPOTT; GUAN [21].

4.4 Case study: Long-term hydrothermal operational planning model

In this section, we present a simplified long-term hydrothermal operational planning model that is the running example in this dissertation. Based on the dynamic programming formulation, we define the long-term hydrothermal operational planning model as:

$$
\begin{aligned}
Q_t(v_t, a_t) = \min_{(v_{t+1}, q_t, s_t, g_t, df_t, f_t)} \quad & c^\top g_t + c_{df}^\top df_t + \beta \overline{\mathcal{Q}}_{t+1}(v_{t+1}) \\
\text{s.t.} \quad & v_{t+1} = v_t + a_t - q_t - s_t, \\
& q_t + M_I g_t + df_t + M_D f_t = d_t, \\
& 0 \le v_{t+1} \le \overline{v}, \quad 0 \le q_t \le \overline{q}, \quad \underline{g} \le g_t \le \overline{g}, \\
& 0 \le f_t \le \overline{f}, \quad 0 \le s_t, \quad 0 \le df_t,
\end{aligned} \tag{4.18}
$$

$$
\overline{\mathcal{Q}}_{t+1}(v_{t+1}) = \begin{cases} \mathbb{E}\left[Q_{t+1}(v_{t+1}, a_{t+1})\right] & , t \in \{1, \ldots, T-1\} \\ 0 & , t = T \end{cases},
$$

for all $t = 1, \ldots, T$. For each stage t, the decision vector is $x_t = (v_{t+1}, q_t, s_t, g_t, df_t, f_t)$, where v_{t+1} is the final stored energy vector of stage t, q_t is the turbined energy vector during stage t, s_t is the spilled energy during stage t, g_t is the thermal generation vector during stage t, df_t is the deficit (load shedding) during stage t, and f_t is the energy interchange vector between subsystems during stage t. In this model, the only uncertainty considered are the energy inflows a_t of each subsystem, which we assume as stagewise independent.

Regarding the constraints, we consider a hydro balance equation, $v_{t+1} = v_t + a_t - q_t - s_t$, and a load balance equation $q_t + M_I g_t + df_t + M_D f_t = d_t$, where d_t is the energy demand vector of each subsystem, M_I is an 0-1 indicator matrix that associates a thermal generation component to the corresponding subsystem, and M_D is a matrix that provides the correct sign for the energy interchange vector f_t, which is $+1$ or -1 if the corresponding component receives or sends energy from or to another subsystem, and 0 if the subsystems are not connected. We also consider upper and lower limits for each variable.

In the objective function, we minimize the sum of thermal generation costs, $c^\top g_t$, deficit costs, $c_{df}^\top df_t$, and the future cost-to-go function, $\overline{\mathcal{Q}}_t(v_{t+1})$, corrected by a discount factor β. We note that the hydro generation has no immediate cost in this model.

In order to emphasize the differences among other formulations, we will represent in section 6.3 the bound constraints by the set $\mathcal{X}_t$ and the decision vector by x_t:

$$x_t := (v_{t+1}, q_t, s_t, g_t, df_t, f_t) \in \mathcal{X}_t,$$

Thus, planning model (4.18) is compactly represented as

$$\begin{aligned} Q_t(v_t, a_t) = \quad \min_{x_t \in \mathcal{X}_t} \quad & c^\top g_t + c_{df}^\top df_t + \beta \overline{\mathcal{Q}}_{t+1}(v_{t+1}) \\ \text{s.t.} \quad & (*)_{\mathrm{P}}, \end{aligned} \tag{4.19}$$

where $(*)_{\mathrm{P}}$ indicates the load and energy balance equations from (4.18). Note that we also omit the definition of the future cost-to-go $\overline{\mathcal{Q}}_{t+1}(v_{t+1})$.

5 DISJUNCTIVE CONSTRAINTS

The technique of Disjunctive Constraints provides a method for describing "if-then" rules in terms of union of polyhedra. There are several important operational planning rules that fits into this category of constraint. For instance, the minimum outflow of a given reservoir defined by the National Water Agency (ANA) considers different values of minimum outflow depending on the stored level situation. It is also possible to represent the risk aversion of low stored volumes by imposing a minimum thermal dispatch if the stored volume is outside a security region, e.g., the one defined by the Risk Aversion Curve (CAR) or the Risk Aversion Surface (SAR). Those examples are described in more detail in section 5.2.

In section 5.1, we present a formula that represents a finite union of polyhedra using 0-1 mixed integer linear constraints. This formula works for a large class of problems, and when it does not work we also guarantee that the corresponding union of polyhedra could not be represented by any collection of linear constraints involving real and binary variables.

5.1 Modeling non-convex constraints with binary variables

Disjunctive programming is optimization over union of polyhedra. While polyhedra are convex, their union do not need to be. In this case, it is impossible to represent such sets using only continuous variables and convex constraints. Let $\{P_i\}_{i \in I}$ be a finite family of polyhedra, where $P_i = \{x \in \mathbb{R}^n \in | \ A_i x \leq b_i\}$, and let P be the corresponding union $\cup_{i \in I} P_i$. The aim of this section is to obtain an algebraic formula that represents P using continuous and binary variables. Let Q

be the following peculiar set:

$$Q = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} x = \sum_{i\in I} x_i,\ A_i x_i \le z_i b_i,\ \sum_{i\in I} z_i = 1, \\ x_i \in \mathbb{R}^n,\ z_i \in \{0,1\},\ i \in I. \end{array} . \right\} \tag{5.1}$$

The set Q plays a central role in understanding how to represent union of polyhedra. We shall prove that, under some regularity conditions, the set Q equals P, and when they are different there is no way to represent P using binary variables and linear constraints.

Lets understand the algebraic representation (5.1). Since each z_i is a binary variable and they sum to one, there is only one $j \in I$ such that z_j equals 1, and then the other z_i's are equal to zero. Note that the recession cone of P_i has the following form:

$$\mathrm{recc}(P_i) = \{d \in \mathbb{R}^n \mid A_i d \le 0\},$$

for all $i \in I$. Then, each vector $x \in Q$ is the sum of a vector $x_j \in P_j$ and of recession directions d_i, for $i \neq j$. This is half the proof of Lemma 5.1 below. To prove the general formula (5.2), we introduce the "algebraic recession cone", defined for a polyhedral set $P = \{x \in \mathbb{R}^n \mid Ax \le b\}$ as $\mathrm{recc}^{\mathrm{A}}(P) = \{d \in \mathbb{R}^n \mid Ad \le 0\}$. Thus, if P is nonempty, the algebraic recession cone of P equals the the recession cone of P, but if P is empty the notion of the algebraic recession cone is still well defined.

Lemma 5.1. *The set Q defined in* (5.1) *has the following expression:*

$$Q = \bigcup_{i\in I} P_i + \sum_{k\in I} \mathrm{recc}^{A}(P_k). \tag{5.2}$$

Proof. Let $\overline{P}$ be the set on the right-hand side of equation (5.2),

$$\overline{P} = \bigcup_{i\in I} P_i + \sum_{k\in I} \mathrm{recc}^{\mathrm{A}}(P_k).$$

We have seen above that Q is a subset of $\overline{P}$. We prove the opposite inclusion. Let x be a vector of $\overline{P}$. Then,

$$x = x_j + \sum_{k \in I} d_k = (x_j + d_j) + \sum_{\substack{k \in I \\ k \neq j}} d_k,$$

for some $x_j \in P_j$ and $d_k \in \mathrm{recc}(P_k)$, where $k \in I$. Note that $x_j + d_j$ belongs to P_j. So, let $\tilde{x}_i$ and $\tilde{z}_i$ be the following variables:

$$\tilde{x}_i = \begin{cases} x_j + d_j & i = j, \\ d_i & i \neq j, \end{cases} \quad \text{e} \quad \tilde{z}_i = \begin{cases} 1 & i = j, \\ 0 & i \neq j, \end{cases}$$

Therefore, $x = \sum_{i \in I} \tilde{x}_i$, $1 = \sum_{i \in I} \tilde{z}_i$, $A\tilde{x}_i \leq \tilde{z}_i b_i$, and $\tilde{z}_i \in \{0, 1\}$, for all $i \in I$. We conclude that x belongs to Q. $\square$

An important remark which follows from Lemma 5.1 is that if all polyhedra have the same recession cone, i.e., $\mathrm{recc}(P_i) = R$, for all $i \in I$, then the algebraic formula (5.1) represents P, that is, $Q = P$. However, a question that arises is: "if Q is different from P, is there another way to represent P using binary variables and linear constraints?" The following theorem shows that if (5.1) does not suffice, no 0-1 linear representation of P exists. This is a result from JEROSLOW [18] whose statement was slightly modified and the corresponding proof simplified.

Theorem 5.2 (Jeroslow). *Suppose that P is described by a system of linear equations with mixed integer variables, i.e., there are matrices A, B, C and a vector b such that*

$$P = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} Ax + By + Cz \leq b, \\ y \in \mathbb{R}^m, \;\; z \in \{0,1\}^l \end{array} \right\}. \tag{5.3}$$

Then, there is a representation of P by a finite union of nonempty polyhedra, $P = \bigcup_{i \in I} P_i$, where all polyhedra have the same recession cone. Moreover, any representation of P by a union of nonempty polyhedra $\bigcup_{i' \in I'} \widetilde{P}_{i'}$ lead to a set $\widetilde{Q}$ that equals P, when using the algebraic formula (5.1).

Proof. Let z_i be a binary vector of $\{0,1\}^l$, and let P_i be the polyhedron defined by setting z equal to z_i in inequality (5.3):

$$P_i := \{x \in \mathbb{R}^n \mid Ax + By \leq b - Cz_i,\ y \in \mathbb{R}^m\}.$$

Let I be the set of indexes i such that P_i is nonempty. From Corollary 2.9 (page 26), the recession cone of these P_i's is given by:

$$R = \{d_x \in \mathbb{R}^n \mid Ad_x + Bd_y \leq 0,\ d_y \in \mathbb{R}^m\}. \tag{5.4}$$

Since this expression is invariant by i, we have proved the first part of the theorem.

Let's prove the second part. Suppose that P could be represented by any finite union of nonempty polyhedra $P = \bigcup_{i' \in I'} \widetilde{P}_{i'}$. From Lemma 5.1 (page 97),

$$\widetilde{Q} = \bigcup_{i' \in I'} \widetilde{P}_{i'} + \sum_{i' \in I'} \operatorname{recc}(\widetilde{P}_{i'}) = P + \sum_{i' \in I'} \operatorname{recc}(\widetilde{P}_{i'}).$$

We claim that $\sum_{i' \in I'} \operatorname{recc}(\widetilde{P}_{i'})$ equals the polyhedral cone R from (5.4). If so, the set $\widetilde{Q}$ would be equal to $P + R$, which in turn is equal to P, since each polyhedron P_i has recession cone equal to R.

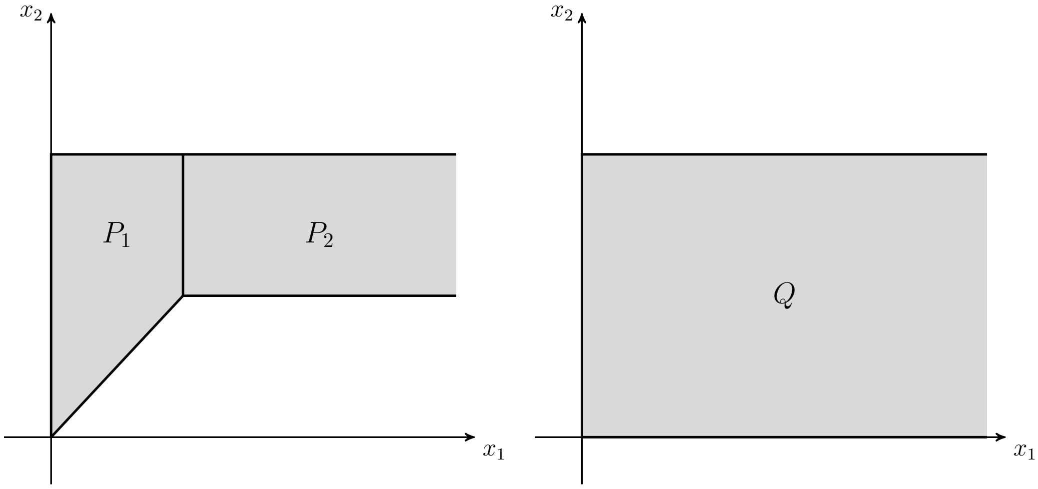
Let's prove that $\sum_{i' \in I'} \operatorname{recc}(\widetilde{P}_{i'}) = R$. Indeed, if P is bounded, the polyhedra P_i and $\widetilde{P}_{i'}$ must be bounded, for all i, i', so the corresponding recession cones are equal to $\{0\}$. In this case, the identity $\sum_{i' \in I'} \operatorname{recc}(\widetilde{P}_{i'}) = R$ is trivially satisfied. If P is an unbounded set, there is a polyhedron $\widetilde{P}_{j'}$ that is unbounded, for some $j' \in I'$, and every polyhedron P_i is unbounded, since they have the same recession cone R. Let d be a nonzero direction of recession of R. If d does not belong to any $\widetilde{P}_{i'}$, then starting from any point $x \in \widetilde{P}_{i'}$ we have that $x + td$ does not belong to $\widetilde{P}_{i'}$ for some large enough scalar $t > 0$. However, the vector x belongs to some polyhedron P_i, and $x + td$ also belongs to P_i, for every positive scalar t, since d is a direction of recession of R. This is a contradiction with the hypothesis that both unions of polyhedra are equal to the same set P. Therefore, R is a subset of $\sum_{i' \in I'} \operatorname{recc}(\widetilde{P}_{i'})$.

For the converse inclusion, let d be a nonzero direction of $\text{recc}(\widetilde{P}_{j'})$, and suppose that d does not belong to R. Let x be a vector of $\widetilde{P}_{j'}$. For a large enough scalar $t > 0$, $x + td$ does not belong to any of these P_i's, but it belongs to $\widetilde{P}_{j'}$. This is again a contradiction with the hypothesis of generating the same set P. Therefore, all recession cones $\text{recc}(\widetilde{P}_{i'})$'s are subsets of R. Since R is a convex cone, the sum of directions of R also belongs to it, so we conclude that $\sum_{i' \in I'} \text{recc}(\widetilde{P}_{i'})$ is a subset of R. $\square$

The hypothesis of nonempty union of polyhedra $\bigcup_{i' I'} \widetilde{P}_{i'}$ in the proof of Theorem 5.2 is necessary, since the formula (5.1) considers each algebraic recession cone $\text{recc}(\widetilde{P}_{i'})$ even if the polyhedron $\widetilde{P}_{i'}$ is empty, as shown in Lemma 5.1. Thus, given a representation of P as union of nonempty polyhedra, we can always aggregate empty polyhedra $\widetilde{P}_{j'}$'s with arbitrary algebraic recession cones that makes the set Q larger than P.

The algebraic formula (5.2) is an important modeling tool, since it provides an explicit description of union of polyhedra using linear constrains and binary variables, and when it fails we have the guarantee that it is impossible to describe that particular case using any other formula. We also note that the formula (5.2) is an extension of the “Big-M” modeling approach which is a mixed integer linear formula for union of bounded polyhedra. If P is bounded, the corresponding polyhedra are bounded, and the recession cones are all equal to $\{0\}$. Therefore, the algebraic formula (5.2) is always able to represent a bounded union P. In figure 5.1a, we show an example of a union of polyhedra that cannot be represented using binary variables, and next to it figure 5.1b depicts the corresponding set Q generated by (5.1).

An useful information for solving mixed integer problems is the knowledge of the convex hull of the corresponding feasible set. We have seen in Lemma 3.7 (page 47) that the infimum of a linear function over a set P is equal to the infimum of



(a) Set not representable by binary variables. (b) Certificate of non-representability.

Figure 5.1: Union of polyhedra not representable by binary variables and linear constraints.

the same function over the convex hull of P. One can easily show that this infimum is also equal when we change P by the closure of the convex hull of P. Let $\overline{Q}$ be the continuous relaxation of formula (5.1):

$$\overline{Q} = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} x = \sum_{i \in I} x_i,\ A_i x_i \leq z_i b_i,\ \sum_{i \in I} z_i = 1, \\ x_i \in \mathbb{R}^n,\ z_i \in [0,1],\ i \in I. \end{array} \right\}. \tag{5.5}$$

The following Theorem states that $\overline{Q}$ is the closure of the convex hull of P. This important idea gave rise to the Lift-and-Project algorithm from Balas (GOMORY [15] and CORNUÉJOLS [13]), which is, along with the Gomory Cuts, the state of the art techniques for solving 0-1 mixed integer linear programs.

Theorem 5.3 (Balas)**.** *Let P be a union of nonempty polyhedra $\bigcup_{i \in I} P_i$. The closure of the convex hull of P,* $\operatorname{cl}\operatorname{conv} P$*, is the continuous relaxation of formula* (5.1)*:*

$$\overline{Q} = \operatorname{cl}\operatorname{conv} P.$$

Proof. Let $x \in \overline{Q}$. Then, there are $x_i \in \mathbb{R}^n$ and $z_i \in [0,1]$ such that

$$x = \textstyle\sum_{i \in I} x_i, \quad A_i x_i \leq z_i b_i, \quad \sum_{i \in I} z_i = 1,$$

for all $i \in I$. Let I^0 and I^+ the set of indexes i such that z_i is equal to zero and greater than zero, respectively. Note that x_i belongs to $\text{recc}(P_i)$, for all $i \in I^0$, and $\widetilde{x}_i = x_i/z_i$ belongs to P_i, for all $i \in I^+$. So, the vector x has the following form:

$$\begin{aligned} x &= \sum_{i \in I^+} z_i \frac{x_i}{z_i} + \sum_{i \in I^0} x_i \\ &= \underbrace{\sum_{i \in I^+} z_i \widetilde{x}_i}_{\in \text{conv}(\bigcup_{i \in I} P)} + \underbrace{\sum_{i \in I^0} x_i}_{\in \sum_{i \in I} \text{recc}(P_i)} . \end{aligned}$$

From Theorem 2.13 (page 33), $\text{cl}\,\text{conv}\, P$ also has the following expression:

$$\text{cl}\,\text{conv} \left(\bigcup_{i \in I} P_i \right) = \text{conv} \left(\bigcup_{i \in I} P_i \right) + \sum_{i \in I} \text{recc}(P_i).$$

Therefore, x belongs to $\text{cl}\,\text{conv}(P)$.

On the other hand, let $x \in \text{cl}\,\text{conv}(P)$. Again from Theorem 2.13,

$$x = \sum_{i \in I} \lambda_i y_i + \sum_{i \in I} d_i,$$

where $y_i \in P_i$, $d_i \in \text{recc}(P_i)$ and $\lambda_i \in [0, 1]$ such that $\sum_{i \in I} \lambda_i = 1$, for all $i \in I$. Let $\overline{I}^0$ and $\overline{I}^+$ be the set of indexes i such that $\lambda_i = 0$ and $\lambda_i > 0$, respectively. Then,

$$x = \sum_{i \in \overline{I}^+} \lambda_i \left(y_i + \frac{1}{\lambda_i} d_i \right) + \sum_{i \in \overline{I}^0} d_i.$$

Let $\widetilde{x}_i$ be equal to $y_i + (1/\lambda_i) d_i$, and note that $\widetilde{x}_i$ belongs to P_i, since y_i belongs to P_i and d_i belongs to $\text{recc}(P_i)$, for all $i \in \overline{I}^+$. Let x_i and z_i be the following vectors:

$$x_i = \begin{cases} \lambda_i \widetilde{x}_i, & \text{se } i \in I^+, \\ d_i & \text{se } i \in I^0, \end{cases} \qquad z_i = \lambda_i.$$

Therefore,

$$x = \textstyle\sum_{i \in I} x_i, \quad A_i x_i \leq z_i b_i, \quad \sum_{i \in I} z_i = 1,$$

and so x belongs to $\overline{Q}$. $\square$

5.2 Some applications of Disjunctive Constraints

Throughout this section we present some applications of Disjunctive Constraints using the notation of section 4.4. Other Brazilian's research in Disjunctive Constraints can be found in CAMPELLO [9], CAMPELLO; FILHO [10] and CAMPELLO; FILHO [11].

In subsections 5.2.1 and 5.2.2, we introduce some operational risk aversion proposals that aim to increase thermal generation if the stored energy is outside a given security region. These are the Binary Risk Aversion Curve (CAR-B) and the Binary Risk Aversion Surface (SAR-B) methodologies.

In subsection 5.2.3, we address the topic of deficit generation. Modeling soft constraints with penalty methods may introduce side effects such as the preventive deficit, i.e., occurrence of load curtailment when there is still a reasonable stored energy available. We propose a non-convex constraint which restricts the possibility of load curtailment to a specific situation of energy stored level.

In subsection 5.2.4, we propose a non-convex constraint for modeling the minimum energy outflow constraint. The minimum outflow depends on the amount of stored energy, and for lower storage levels the required minimum outflow becomes smaller. This is a nontrivial example of application of the formula 5.1, since, without it, obtaining an expression that represents such constraint would be considerably difficult.

5.2.1 Binary Risk Aversion Curve – CAR-B

The Risk Aversion Curve (CAR) is a traditional methodology in the Brazilian Power System to determine a security storage region. On the basis of certain assumptions, a safety storage curve is established for each subsystem, and having the stored energy above the given curve means that the system is in the safe region of storage.

In the *Binary Risk Aversion Curve* (CAR-B), if the stored level is below the CAR, then a minimum amount of thermal generation is compulsorily dispatched. At a given stage t, we denote by v_{MinOp} the vector corresponding to the minimum security energy level of each subsystem, and by $\underline{G}$ the vector of thermal security dispatch of each thermal plant. We omit the subscript t of the stage to not overload the notation. Thus, the constraints corresponding to the Binary Risk Aversion Curve approach is given by

$$\begin{array}{rcl} v_{t+1} & \geq & (1 - z_g) \cdot v_{\text{MinOp}}, \\ g_t & \geq & z_g \cdot \underline{G}, \\ z_g & \in & \{0, 1\}, \end{array} \tag{5.6}$$

where z_g is a binary variable that indicates if the final stored energy v_{t+1} is below the security level v_{MinOp}. If so, the thermal generation lower bound is raised to a level $\underline{G}$, see figure 5.2. A straightforward generalization of this idea is to consider one binary variable for each subsystem, so that if the stored energy of a subsystem is below the corresponding target just a group of thermal plants is dispatched, instead of all them. Note that the algebraic equation (5.6) represents the union of the following polyhedra:

$$P_1 = \{(v_{t+1}, g_t) \mid v_{\text{MinOp}} \leq v_{t+1} \leq \overline{v},\ 0 \leq g_t \leq \overline{g}\},$$
$$P_2 = \{(v_{t+1}, g_t) \mid 0 \leq v_{t+1} \leq \overline{v},\ \underline{G} \leq g_t \leq \overline{g}\}.$$

and using the formula (5.1) on $P_1 \cup P_2$ we get (5.6).

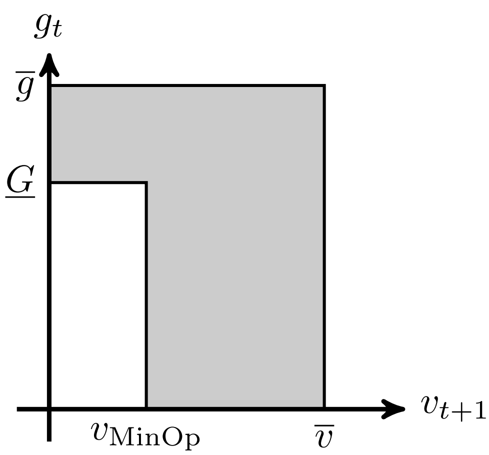


Figure 5.2: Binary Risk Aversion Curve (CAR-B).

5.2.2 Binary Risk Aversion Surface – SAR-B

The Risk Aversion Surface (SAR) is an extension of the CAR approach, since it considers the possibility of energy interchanges between subsystems when describing the energy safety region. In the CAR approach, a fixed energy interchange vector is imposed to simplify the computations and to have uncoupled security energy levels between the subsystems. In the SAR approach, the energy interchange couples the safe storage states and makes the explicit computation of the operational safety region more involved. In practice, the SAR region is described by a convex optimal value function $S_{t+1}(\cdot)$. In this case, a stored energy vector v_{t+1} belongs to the safety region if and only if $S_{t+1}(v_{t+1}) \leq 0$, see figure 5.3. The SAR function also has a straightforward upper bound, denoted by M (Big-M), which makes possible the binary modeling approach.

A natural extension of the CAR-B is the *Binary Risk Aversion Surface* (SAR-B). We just slightly change formulation (5.6), using the SAR function $S_{t+1}(\cdot)$:

$$\begin{array}{rcl} S_{t+1}(v_{t+1}) & \leq & z_g \cdot M, \\ g_t & \geq & z_g \cdot \underline{G}, \\ z_g & \in & \{0, 1\}. \end{array} \tag{5.7}$$

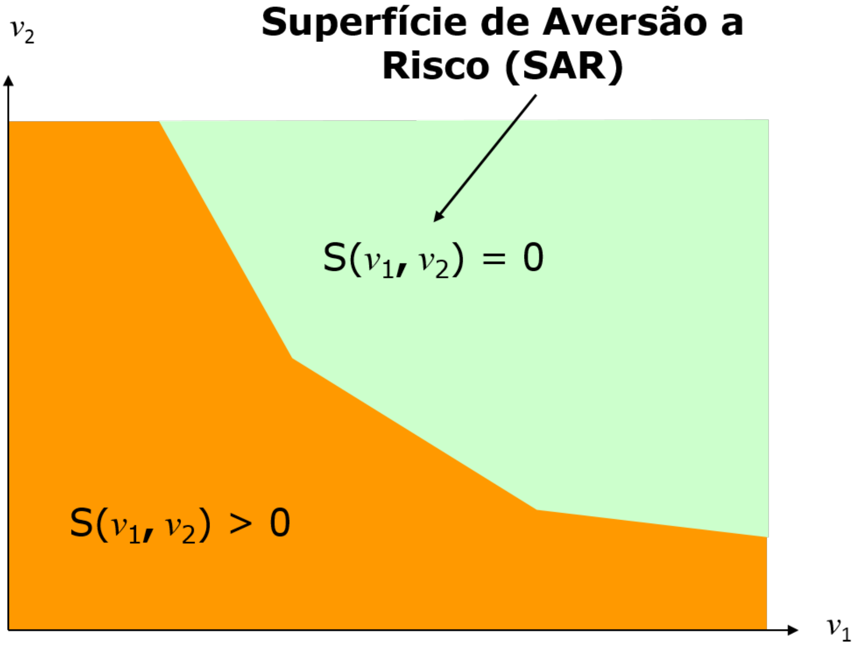


Figure 5.3: SAR region

A generalization of the SAR-B to consider a localized thermal dispatch is much more involved, since the SAR function is positive if the stored energy vector is outside the safety region, but it does not indicate which subsystem is depleted. The SAR-B constraints induce the following polyhedra:

$$P_1 = \{(v_{t+1}, g_t) \mid S_{t+1}(v_{t+1}) \leq 0,\ 0 \leq g_t \leq \overline{g}\},$$
$$P_2 = \{(v_{t+1}, g_t) \mid S_{t+1}(v_{t+1}) \leq M,\ \underline{G} \leq g_t \leq \overline{g}\}.$$

It is instructive to note that $S_{t+1}(v_{t+1}) \leq M$ is equivalent to $0 \leq v_{t+1} \leq \overline{v}$, since M is the upper bound of the function $S_{t+1}(\cdot)$ and does not impose any restriction on the stored volume v_{t+1}.

5.2.3 Suppression of preventive deficit (SPDef)

Load shedding is an important variable for the load balance equation, and its cost also has an economic sense. However, the use of artificial penalties for modeling operational constraints can induce preventive deficit, i.e., load shedding while having a reasonable amount of stored energy available. This happens because the penalty distorts the future cost function, which causes a significant increase in the water value. One modeling approach to avoid this side-effect is the *Suppression of preventive deficit* (SPDef):

$$\begin{array}{rcccl} 0 & \leq & v_{t+1} & \leq & (1-z_d)\cdot\overline{v} + z_d v_{\text{def}}, \\ 0 & \leq & df_t & \leq & z_d \cdot d_t, \\ & & z_d \in \{0,1\}, & & \end{array} \tag{5.8}$$

where z_d is a binary variable that indicates if the final stored energy v_{t+1} is below the deficit point v_{def}. In this approach, load shedding is allowed only if the final stored energy v_{t+1} is below a certain storage value v_{def}, and the load shedding cannot exceed the demand d_t, see figure 5.4. This last constraint turns the union of polyhedra a bounded set, which makes possible the representation of that set using formula (5.1). Note that the SPDef approach represents the union of the following polyhedra:

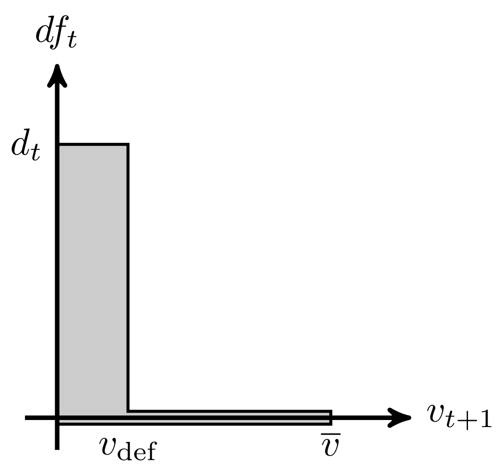


Figure 5.4: Suppression of Preventive Deficit (SPDef).

$$P_1 = \{(v_{t+1}, df_t) \mid 0 \le v_{t+1} \le \overline{v},\ df_t = 0\},$$
$$P_2 = \{(v_{t+1}, df_t) \mid 0 \le v_{t+1} \le v_{\text{def}},\ 0 \le df_t \le d_t\}.$$

and using the formula (5.1) on $P_1 \cup P_2$ we get (5.8)

5.2.4 Binary Minimum Outflow – QMinB

The Minimum Outflow constraint represents the minimum water flow rate that ensures the multiple uses of the water for irrigation, industrial activities, human supply, fishing, navigation, recreation, and others. In shortage situations, the priority use of water resources is for human consumption and watering livestock. The *Binary Minimum Outflow* (QMinB) is a proposal that considers the possibility of relaxing the Minimum Outflow parameters in low storage situations to guarantee the continuity of supply for essential humans activities. Denote by $\underline{q}_1$ the minimum outflow in normal circumstances, $\underline{q}_2$ a smaller outflow value for shortage situations, and by v_{crit} the critical storage below which we consider an outflow of $\underline{q}_2$, instead of $\underline{q}_1$. For stored values v_t above v_{crit}, the parameter $\underline{q}_1$ is just a lower bound for the outflow q_t, but for stored values between $\underline{q}_2$ and v_{crit} the corresponding outflow is exactly $\underline{q}_2$, and for stored values below $\underline{q}_2$ the outflow is the remaining stored volume v_t. Thus, the Binary Minimum Outflow describes a feasible set for the outflow and the initial stored energy pair (q_t, v_t) that is the union of the following polyhedra:

$$P_1 = \{(q_t, v_t) \mid v_{\text{crit}} \le v_t \le \overline{v},\ \underline{q}_1 \le q_t \le \overline{q}\},$$
$$P_2 = \{(q_t, v_t) \mid \underline{q}_2 \le v_t \le v_{\text{crit}},\ q_t = \underline{q}_2\},$$
$$P_3 = \{(q_t, v_t) \mid 0 \le v_t \le \underline{q}_2,\ q_t - v_t = 0\},$$

see figure 5.5 for an illustration.

The description of the above union of polyhedra using binary variables may seem complicated at first but, in fact, the algebraic formula (5.1) provides an al-

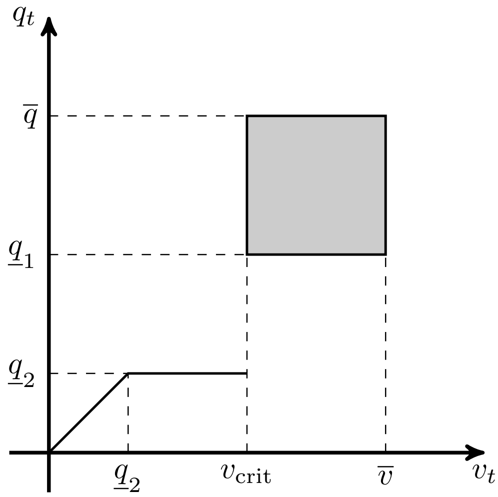


Figure 5.5: Binary Minimum Outflow (QMinB).

gorithmic way to represent it. We have to create one binary variable $z_i \in \{0,1\}$ and one pair (q_t^i, v_t^i) for each polyhedron P_i, so that (q_t^i, v_t^i) satisfies the linear constraints of P_i with the binary variable z_i multiplying the right-hand side, and the binary variables z_i sum up to one, i.e., $z_1 + z_2 + z_3 = 1$, where $i = 1, 2, 3$. Denote by P the union of polyhedra $P_1 \cup P_2 \cup P_3$. Thus, the vector (v_t, q_t) defined by the sum $(v_t^1, q_t^1) + (v_t^2, q_t^2) + (v_t^3, q_t^3)$ represents the union of polyhedra P:

$$P = \left\{ (q_t, v_t) \;\middle|\; \begin{array}{c} q_t = q_t^1 + q_t^2 + q_t^3, \qquad z_1 + z_2 + z_3 = 1, \\ v_t = v_t^1 + v_t^2 + v_t^3, \quad z_i \in \{0,1\}, \ i = 1,2,3, \\ \\ z_1 v_{\text{crit}} \le v_t^1 \le z_1 \overline{v}, \quad z_2 \underline{q}_2 \le v_t^2 \le z_2 v_{\text{crit}}, \quad 0 \le v_t^3 \le z_3 \underline{q}_2, \\ z_1 \underline{q}_1 \le q_t^1 \le z_1 \overline{q}, \qquad q_t^2 = z_2 \underline{q}_2, \qquad q_t^3 - v_t^3 = 0. \end{array} \right\}.$$

6 STOCHASTIC DUAL DYNAMIC *INTEGER* PROGRAMMING

6.1 Local, Benders, Strengthened Benders and Lagrangian cuts

In this section we discuss several types of cuts for a particular class of non-convex optimal value function: the optimal value function of a 0-1 MILP. The results of this section may be extended to general MILPs, but for the sake of exposition, we adopt this particular approach. We rely on the following optimal value function to present the various types of cuts:

$$\begin{array}{rl} f(b) = \min_{x,z} & c^\top x + q^\top z \\ \text{s.t.} & Ax + Gz \leq b, \\ & (x,z) \in X, \end{array} \tag{6.1}$$

where $X = \left\{(x,z) \in \mathbb{R}^n \times \{0,1\}^l \;\middle|\; A'x + G'z \leq b'\right\}$. Note that the right-hand side b' is fixed, so we only assess the variability of f with respect to b.

The first cut notion that we present is the *local cut*. The local cut is a hyperplane that supports f in a neighborhood of a given point $(\bar{b}, f(\bar{b}))$, see figure 6.2. More precisely, we say that $g \in \mathbb{R}^m$ is a *local subgradient* of f at $\bar{b}$ if

$$f(b) \geq f(\bar{b}) + g^\top (b - \bar{b}), \quad \forall b \in B_\delta(\bar{b}),$$

where $B_\delta(\bar{b})$ is an open ball centered on $\bar{b}$ with some radius $\delta > 0$. The function f is *locally subdifferentiable* at $\bar{b}$ if the set of corresponding local subgradient, denoted by $\partial f(b)$, is nonempty. The *local cut* of f at $\bar{b}$ is the affine function $h(b) = f(\bar{b}) + g^\top (b - \bar{b})$, where g is a local subgradient at $\bar{b}$.

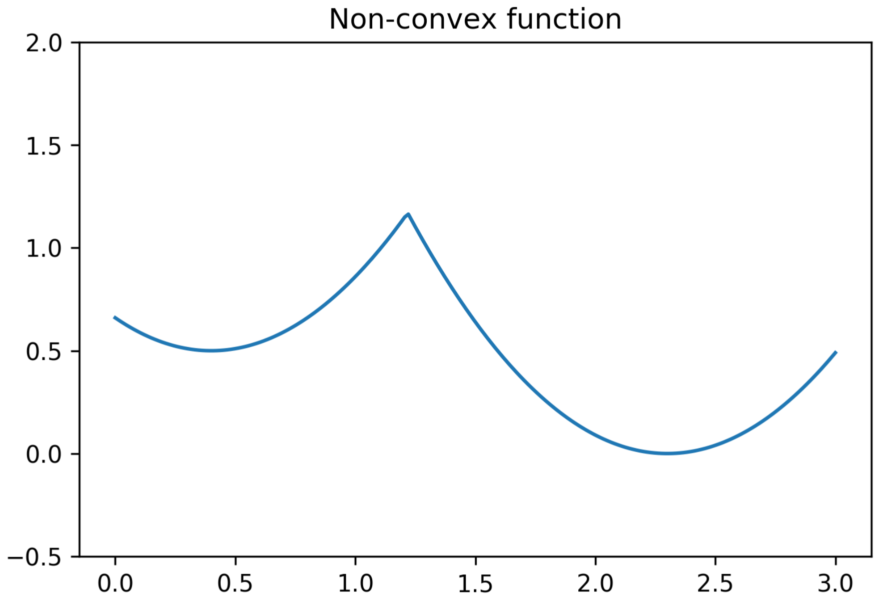


Figure 6.1: Pictorial illustration of a 0-1 MILP optimal value function.

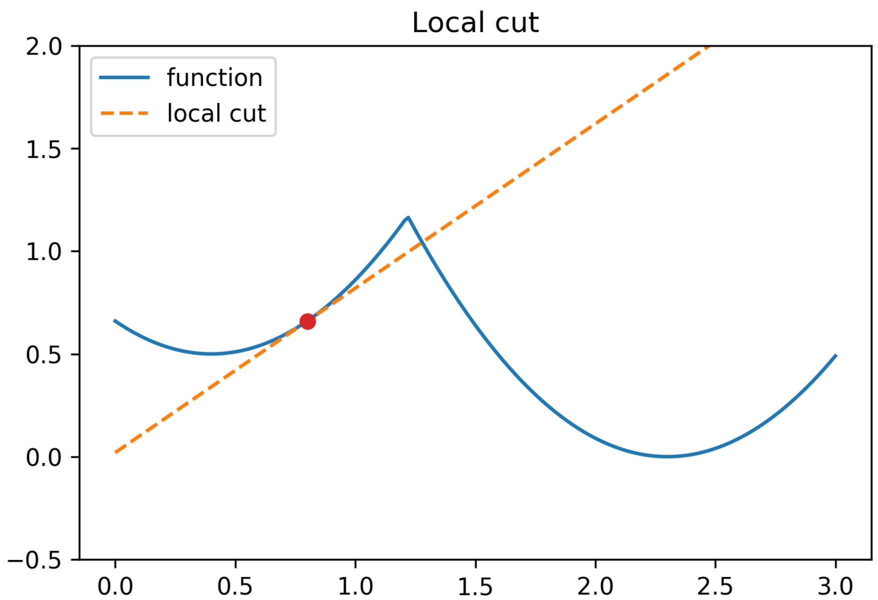


Figure 6.2: Local cut

A procedure to obtain a local subgradient for f is based on solving (6.1) to optimality and computing the dual optimal solution of the LP generated by assigning to z the binary optimal component z^* in problem (6.1). Let us denote by f_z the optimal value function of problem (6.1) with the binary variable z fixed:

$$\begin{aligned} f_z(b) = q^\top z + \min_x \quad & c^\top x \\ \text{s.t.} \quad & Ax \le b - Gz, \\ & (x, z) \in X. \end{aligned}$$

The following theorem states a sufficient condition in which this procedure results into a local subgradient for f.

Theorem 6.1 (Local subdifferentiability of a 0-1 MILP)**.** *If the optimal binary component z^* of* (6.1) *is unique for a given right-hand side b, then the optimal value function* (6.1) *is locally subdifferentiable with $\partial f_{z^*}(b) \subset \partial f(b)$.*

Proof. The optimal integer solution z^* is unique if and only if $f_{z^*}(b) < f_z(b)$, for every $z \in \{0,1\}^l$ different from z^*. Let $h_{z^*}(\tilde{b}) = f_{z^*}(b) + g_{z^*}^\top(\tilde{b} - b)$, where g_{z^*} is a subgradient of f_{z^*} at b. Note that $h_{z^*}(b) < f_z(b)$, for every $z \in \{0,1\}^l$ different from z^*. Since each f_z is continuous in $\text{dom}(f_z)$ and there is a finite number of functions f_z's, the following inequality holds:

$$h_{z^*}(\tilde{b}) < f_z(\tilde{b}), \qquad \forall z \neq z^*, \ \forall \tilde{b} \in B_\delta(b),$$

for some $\delta > 0$, where $B_\delta(b)$ is an open ball centered at b with radius δ. Thus, the vector $g_{z^*} \in \mathbb{R}^m$ is a local subgradient of f. □

The notion of local cut is not suitable for using along with traditional algorithms for large-scale multistage stochastic programs such as the SDDP and Nested Cutting Plane. However, this lead to a suitable notion of marginal cost for the MILP problems, see Wolsey et al.

A widely used type of cut that is also applicable to MILP problems is the *Benders cut.* The Benders cut is a supporting hyperplane of the optimal value function f^{LP} defined from the linear programming relaxation of (6.1). Indeed, let X_{LP} be the set given by the continuous relaxation of the binary variable z,

$$X_{\mathrm{LP}} = \left\{(x, z) \in \mathbb{R}^n \times [0,1]^l \;\middle|\; A'x + G'z \leq b'\right\},$$

and let f^{LP} be the following optimal value function:

$$\begin{array}{rl} f^{\mathrm{LP}}(b) = \min_{x,z} & c^\top x + q^\top z \\ \text{s.t.} & Ax + Gz \leq b, \\ & (x, z) \in X_{\mathrm{LP}}. \end{array} \tag{6.2}$$

Since X_{LP} contains X, the function f^{LP} is less than or equal to f for every right-hand side b. Therefore,

$$f(b) \geq f^{\mathrm{LP}}(b) \geq f^{\mathrm{LP}}(\bar{b}) + g_{\mathrm{LP}}^\top(b - \bar{b}), \qquad \forall b \in \mathbb{R}^m,$$

where g_{LP} is a subgradient of f^{LP} at $\bar{b}$, see figure 6.3. The *Benders cut* of f at $\bar{b}$ is the affine function $h(b) = f^{\mathrm{LP}}(\bar{b}) + g_{\mathrm{LP}}^\top(b - \bar{b})$, where g_{LP} is a subgradient of f^{LP} at $\bar{b}$. Note that the linear programming relaxation may be a loose approximation of the corresponding 0-1 MILP problem, and so may be the corresponding Benders cut. Thus, the Benders cut provides a lower bound to the optimal value of a multistage mixed integer stochastic problem, but it may never converge to the optimal policy.

Lagrangian cuts are the tightest type of cut for a nonconvex function f, since they are supporting hyperplanes associated to the convex regularization f^{**} which is the largest convex function less than or equal to f at all points, see section 3.5 for a discussion. As we have seen in Corollary 3.18 (page 68), the convex regularization of the optimal value function f is obtained by replacing the set X from (6.1) by the corresponding convex hull:

$$\begin{array}{rl} f^{**}(b) = \min_{x,z} & c^\top x + q^\top z \\ \text{s.t.} & Ax + Gz \leq b, \\ & (x, z) \in \operatorname{conv}(X). \end{array} \tag{6.3}$$

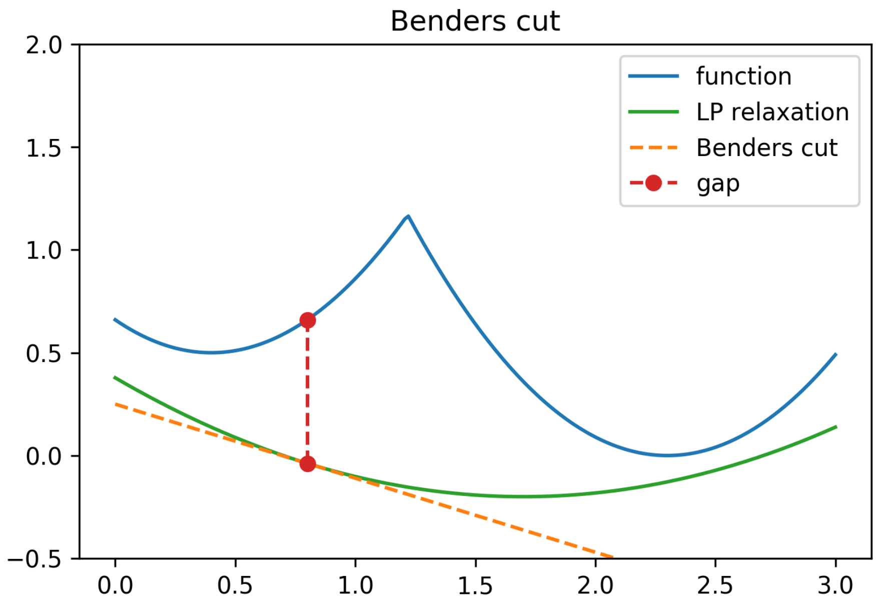


Figure 6.3: Benders cut

Since the convex hull of X is a polyhedron[1], the problem (6.3) is a LP. Thus, the *Lagrangian cut* of f is the affine function $f^{**}(\bar{b}) + g_c^\top(b - \bar{b})$, where g_c is subgradient of f^{**} at $\bar{b}$, see figure 6.4. Note that

$$f(b) \geq f^{**}(b) \geq f^{**}(\bar{b}) + g_c^\top(b - \bar{b}), \qquad \forall b \in \mathbb{R}^m.$$

The PhD thesis of THOMÉ [27] provides an application of the Lagrangian cuts to the cost-to-go functions of a stochastic MILP program for solving a power system planning model.

Both the Benders and Lagrangian cuts were built based on convex approximations of the original 0-1 MILP problem (6.1). These approximations were obtained by replacing the set X by a convex set that contains X. From the subset relation $X \subset \text{conv}(X) \subset X_{\text{LP}}$, we conclude that $f(b) \geq f^{**}(b) \geq f^{\text{LP}}(b)$, for all $b \in \mathbb{R}^m$.

[1] The general MILP case requires further hypothesis about A and B to guarantee that $\text{conv}(X)$ is a polyhedron, see Theorem 3.8 page 49

Therefore, the Lagrangian cuts are tighter than the Benders cut at the point $\bar{b}$ in which they are computed.

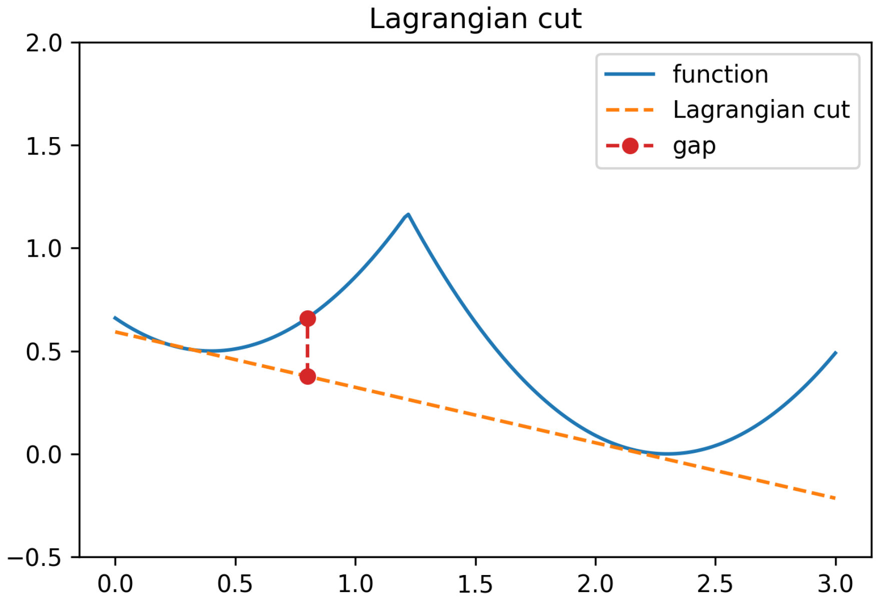


Figure 6.4: Lagrangian cut

In practice, finding the convex hull of X may be difficult, and an iterative process must be performed to calculate the Lagrangian cuts. In order to clarify this procedure and connect the ideas of this document with the SDDiP article, we recall the relationship between the biconjugate of f and the Lagrangian Relaxation of problem (6.1), but starting from the identity (6.3). Indeed, note that

$$
\begin{aligned}
f^{**}(b) &= \max_{\mu\geq 0}\left[\min_{(x,z)\in\mathrm{conv}(X)}\left\{c^\top x + q^\top z + \mu^\top(Ax+Gz-b)\right\}\right] \\
&= \max_{\mu\geq 0}\left[\min_{(x,z)\in X}\left\{c^\top x + q^\top z + \mu^\top(Ax+Gz-b)\right\}\right] \\
&= \max_{g\leq 0}\left[\min_{(x,z)\in X}\left\{c^\top x + q^\top z - g^\top(Ax+Gz-b)\right\}\right] \\
&= \max_{g\leq 0}\left[g^\top b + \min_{(x,z)\in X}\left\{c^\top x + q^\top z - g^\top(Ax+Gz)\right\}\right] \\
&= \max_{g\leq 0}\left[g^\top b + \mathcal{L}(g)\right], \qquad (6.4)
\end{aligned}
$$

where the first identity follows from the strong duality in linear programming, the second one follows from lemma 3.7 (page 47), the third one results from a change of variable from $\mu \geq 0$ to $g \leq 0$, the forth identity results from moving the term $g^\top b$ outside the minimum in (x, y), and the fifty identity follows from denoting by $\mathcal{L}(g)$ the optimal value $\min_{(x,z)\in X}\left\{c^\top x + q^\top z - g^\top(Ax+Gz)\right\}$. The problem (6.4) is the *Lagrangian Relaxation* of (6.1).

It is instructive to note that $\mathcal{L}(g)$ is the linear coefficient of the supporting hiperplane of f with normal vector g. Indeed,

$$
\begin{aligned}
\mathcal{L}(g) &= \min_{\substack{Ax+Gz\leq d,\\ (x,z)\in X,\ d\in\mathbb{R}^m}}\left\{c^\top x + q^\top z - g^\top d\right\} \qquad (6.5) \\
&= \min_{d\in\mathbb{R}^m}\left[\min_{\substack{Ax+Gz\leq d,\\ (x,z)\in X}}\left\{c^\top x + q^\top z - g^\top d\right\}\right] \\
&= \min_{d\in\mathbb{R}^m}\left\{f(d) - g^\top d\right\},
\end{aligned}
$$

where the first identity follows from $Ax + Gz \leq d$ and the hypothesis of $-g \geq 0$, the second identity results from the Benders decomposition, and the third identity follows from the definition (6.1) of $f(d)$. Recall from section 3.5 (page 69) that the optimal value $\min_{d\in\mathbb{R}^m}\left\{f(d) - g^\top d\right\}$ is the linear coefficient of the supporting hyperplane of f with normal vector $g \in \mathbb{R}^m$. Let $r_g(b)$ be such supporting hyperplane, that is, $r_g(b) = g^\top b + \mathcal{L}(g)$. From equation (6.4), we note that f^{**} is the pointwise maximum of $r_g(b)$ over all possible normal vectors $g \leq 0$:

$$f^{**}(b) = \max_{g \leq 0} r_g(b). \tag{6.6}$$

Therefore, the biconjugate of f evaluated at b can be obtained by solving the La-

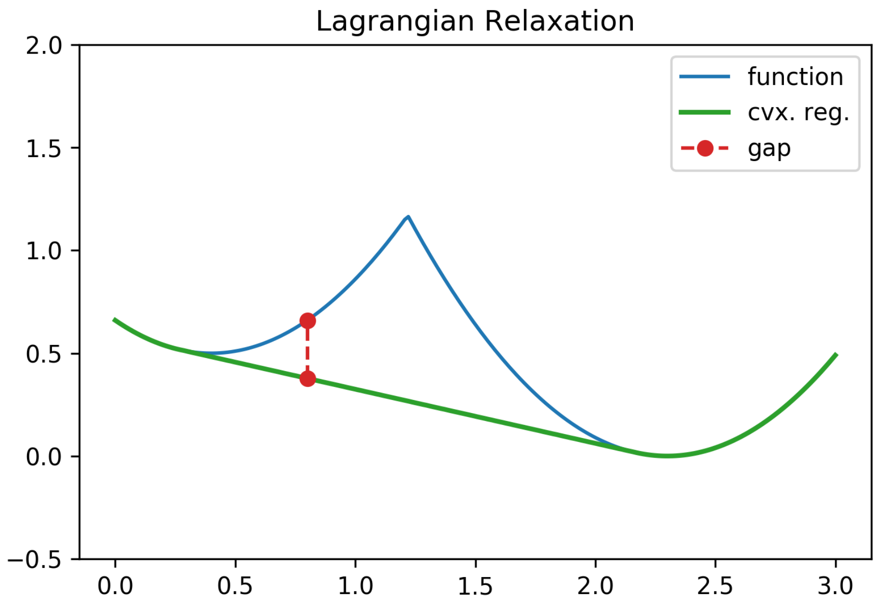


Figure 6.5: Lagrangian Relaxation

grangian Relaxation (6.4) which is equivalent to solving (6.6). Note that the optimal affine function $r_{g^*}(b)$ is the supporting hyperplane of f at b and the corresponding optimal solution g^* is a subgradient of f^{**} at b. The optimal solution of (6.6) can be find using any non-differentiable convex optimization algorithm such as the subgradient method, cutting planes, or bundle method. However, any of those methods requires the evaluations of $r_g(b)$ for different vectors g, whose computational burden is solving the 0-1 MILP problem (6.4) that defines $\mathcal{L}(g)$ for each g. Note that for any fixed $g \leq 0$, the function $r_g(b)$ is a hyperplane less than or equal to f at every point. So, stopping the algorithm at any iteration lead to a lower approximation of

f, but if the solution g is far from optimal the corresponding approximation may be very loose at b.

The last type of cut of this section is the *Strengthened Benders* cut. For a given vector $\bar{b}$, the Strengthened Bender cut at $\bar{b}$ is the affine function $r_{g_{\mathrm{LP}}}(b) = g_{\mathrm{LP}}^{\top} b + \mathcal{L}(g_{\mathrm{LP}})$, where g_{LP} is a subgradient of f^{LP} at $\bar{b}$, see figure 6.6. In order to compute this type of cut, we have to find the dual optimal solution g_{LP} of the LP relaxation (6.2) and evaluate $\mathcal{L}(g_{\mathrm{LP}})$, which is the optimal value of the 0-1 MILP problem (6.5) with $g = g_{\mathrm{LP}}$. Thus, the Strengthened Benders cut is numerically cheaper than the Lagrangian cut. Another important fact is that the Strengthened Benders cut is parallel to the corresponding Benders cut, but tighter than it. This follows from the fact that $f^{\mathrm{LP}}(b)$ is less than or equal to $f(b)$ at every point b and that the associated linear coefficient of the Benders and Strengthened Benders cut are $\min_{d \in \mathbb{R}^m} \left\{ f^{\mathrm{LP}}(d) - g_{\mathrm{LP}}^{\top} d \right\}$ and $\min_{d \in \mathbb{R}^m} \left\{ f(d) - g_{\mathrm{LP}}^{\top} d \right\}$, respectively. So, the linear coefficient of the Benders cut is smaller than the one of Strengthened Benders cut. For a summary of all types of cuts presented in this section, see figure 6.7.

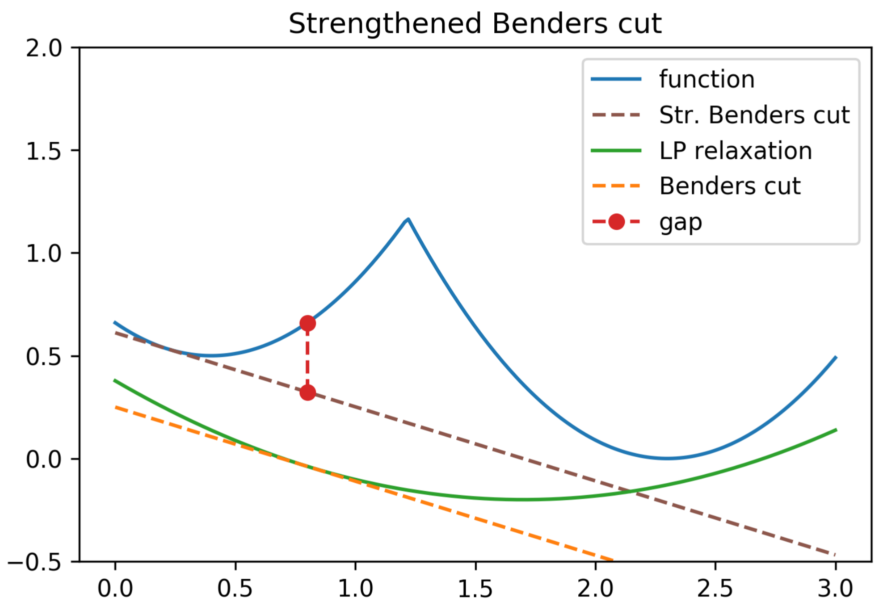


Figure 6.6: Strengthened Benders cut

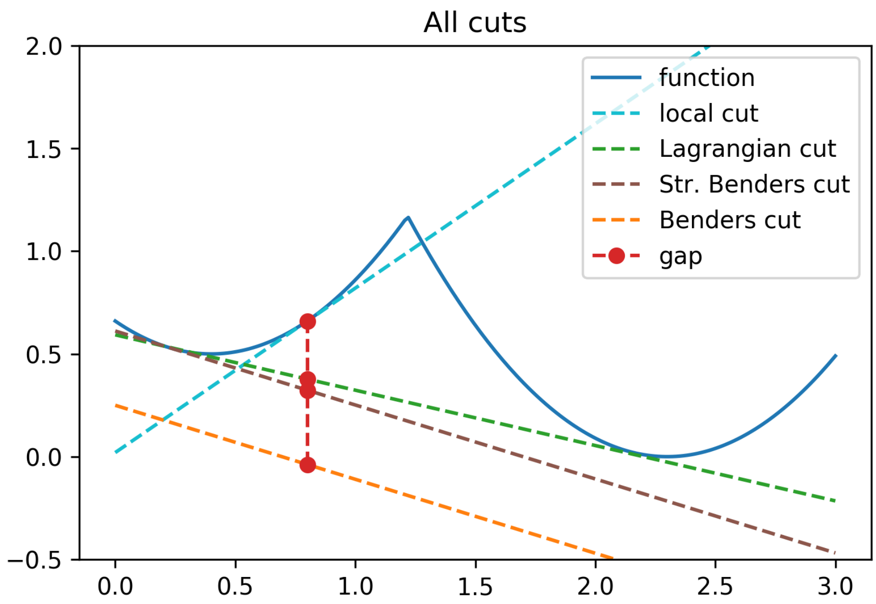


Figure 6.7: All cuts

6.2 SDDiP and the Blessing of Binary

In this section, we present the main result of the SDDiP article, the "Blessing of Binary", which states that the Lagrangian cuts are "tight" (no convexification gap) at some special points for a particular class of 0-1 MILP optimal value functions. Then, we describe how to approximate a general 0-1 MILP problem by a problem of the Blessing of Binary class, and how to bound the corresponding approximation error. Those ideas are also suggestions of the SDDiP article.

The class of optimal value functions used in the Blessing of Binary theorem has the following form:

$$\begin{aligned} \phi(\lambda) = \min_{x,y,z} \quad & c^\top x + \tilde{c}^\top y + q^\top z \\ \text{s.t.} \quad & y = \lambda, \\ & (x, y, z) \in \tilde{X}, \end{aligned} \tag{6.7}$$

where x is a continuous vector of $\mathbb{R}^n$, y is a continuous vector on the unit cube $[0,1]^p$, and z is a binary vector of $\{0,1\}^l$ such that the feasible set $\tilde{X}$ is given by

$$\tilde{X} = \{(x, y, z) \in \mathbb{R}^n \times [0,1]^p \times \{0,1\}^l \mid \tilde{A}x + \tilde{F}y + \tilde{G}z \leq \tilde{b}\}.$$

We denote by ϕ instead of f the optimal value function and by λ instead of b the corresponding right-hand side in (6.7) to emphasize the differences between both classes of functions. It is instructive to note two main restrictive aspects in problem (6.7): the constraint of the right-hand side λ is a trivial equality constraint, i.e., $y = \lambda$, and the variable y is restricted to the unit cube $[0,1]^p$, which means that the problem (6.7) is infeasible if λ do not belong to $[0,1]^p$.

As we have seen in section 6.1, the Lagrangian cut is a supporting hyperplane to the biconjugate ϕ^{**}, which has the following expression:

$$\begin{aligned} \phi^{**}(\lambda) = \min_{x,y,z} \quad & c^\top x + \tilde{c}^\top y + q^\top z \\ \text{s.t.} \quad & y = \lambda, \\ & (x, y, z) \in \operatorname{conv}(\tilde{X}), \end{aligned}$$

according to Corollary 3.18 (page 68). Surprisingly, the Blessing of Binary states that there is no gap between the convex regularization ϕ^{**} and the original function ϕ at the binary points $\hat{\lambda} \in \{0,1\}^p$.

Theorem 6.2 (Blessing of Binary)**.** *The convex regularization ϕ^{**} equals the original function ϕ at the binary points:*

$$\phi^{**}(\hat{\lambda}) = \phi(\hat{\lambda}), \quad \forall \hat{\lambda} \in \{0,1\}^l. \tag{6.8}$$

Proof. Let $\hat{\lambda} \in \{0,1\}^p$. If $\operatorname{conv}(\tilde{X}) \cap (y = \hat{\lambda})$ is empty, then both $\phi^{**}(\hat{\lambda})$ and $\phi(\hat{\lambda})$ are equal to $+\infty$. Assume that $\operatorname{conv}(\tilde{X}) \cap (y = \hat{\lambda})$ is nonempty, and let (x, y, z) be a vector on this set. Then, $y = \hat{\lambda}$ and

$$(x, y, \hat{\lambda}) = \sum_{i=1}^{k} \alpha_i (x_i, y_i, z_i), \tag{6.9}$$

where $(x_i, y_i, z_i) \in \tilde{X}$ and $\alpha_i \in [0,1]$ such that $\sum_{i=1}^{k} \alpha_i = 1$. Without loss of generality, suppose that all α_i's are positives. Since, $\hat{\lambda}$ is an extreme point of the unit cube $[0,1]^p$ and each y_i belongs to $[0,1]^p$, we have that $y_i = \hat{b}$, for all $i = 1, \ldots, k$. Moreover, there is $(x_i, y_i, \hat{\lambda}) \in \tilde{X}$ such that

$$c^\top x_i + \tilde{c}^\top y_i + q^\top \hat{\lambda} \leq c^\top x + \tilde{c}^\top y + q^\top \hat{\lambda},$$

otherwise we have a contradiction with equation (6.9). Thus, we conclude (6.8). □

Motivated by Theorem 6.2, the SDDiP algorithm requires two additional steps before computing the Lagrangian cuts for cost-to-go functions of a stochastic MILP program: the binarization of the state variables, and the dimensional lifting of the feature space. After these two steps, we get a cost-to-go function that fits the hypothesis of Theorem 6.2, so we guarantee that the Lagrangian cuts are tight at the binary coordinates. We can increase the overall quality of the resulting cost-to-go approximation by increasing the number of points in the binarization step. In this section, the cost-to-go function of a stochastic MILP program is represented by the optimal value function f, see (6.1). An additional hypothesis required for the SDDiP algorithm is the boundedness of $\mathrm{dom}(f)$, i.e., $\mathrm{dom}(f) \subset [-U, U]^m$, for some $U > 0$.

The binarization step is the process of approximating the block $[-U, U]^m$ by a uniform grid R, where we have a simple formula involving binary variables to represent the grid elements. We illustrate this idea with a finite interval $[0, U]$. Let $\epsilon > 0$ be such that $\epsilon = U/2^k$, for some $k \in \mathbb{Z}_+$. The uniform grid of $[0, U]$ with precision ϵ is the following set

$$R := \{0, \epsilon, 2\epsilon, 3\epsilon, \ldots, (2^k - 1)\epsilon\}.$$

Note that $[0, U]$ contains R, since all elements of R are non-negative and the largest one is $(2^k - 1)\epsilon = (1 - 2^{-k})U$, which is less than U. Denote by $[k]$ the set of positive integers from 1 to k, i.e., $[k] = \{1, \ldots, k\}$. We can use the following algebraic formula and k binary variables to generate the elements of R:

$$b_R = \sum_{j=1}^{k} \epsilon 2^{j-1} \gamma_j,$$

where $\gamma_j \in \{0, 1\}$, for all $j \in [k]$. If the scalar γ_j can have any value in the interval $[0, 1]$, we can represent any number of $[0, (1 - 2^{-k})U]$.

We shall extend those ideas to a uniform grid on a symmetric block $[-U, U]^m$. The uniform grid with precision $\epsilon = U/2^k$ on $[-U, U]^m$ is the Cartesian Product R^m, where

$$R := \{-(2^k - 1)\epsilon, \ldots, -2\epsilon, -\epsilon, 0, \epsilon, 2\epsilon, \ldots, (2^k - 1)\epsilon\}.$$

Following the same analogy, we represent the elements of R^m using the expression below:

$$b_{R,i} = \sum_{j=1}^{k} \epsilon 2^{j-1} (\gamma_{i,j} - \gamma_{i,j+k}), \quad i \in [m], \tag{6.10}$$

where $\gamma_{i,j} \in \{0,1\}$ for all $i \in [m]$ and $j \in [2k]$. If each $\gamma_{i,j}$ assumes any value along the interval $[0,1]$, the formula (6.10) generates the set $[-(1-2^{-k})U, (1-2^{-k})U]^m$.

In order to clarify the exposition of the binarization step, it is convenient to represent equation (6.10) using a vectorial notation. Let y be a vector of the unit cube $[0,1]^p$ such that $p = 2mk$ and y satisfies to

$$y_{2k(i-1)+j} = \gamma_{i,j},$$

where $i \in [m]$ and $j \in [2k]$. Let $F \in \mathbb{R}^{m \times p}$ be a matrix whose entries $F_{i,r}$ are defined by

$$F_{i,r} = \begin{cases} \epsilon 2^{j-1} & \text{, if } r = 2k(i-1)+j, \text{ for some } j \in [k], \\ -\epsilon 2^{j-1} & \text{, if } r = 2k(i-1)+j+k, \text{ for some } j \in [k], \\ 0 & \text{, otherwise.} \end{cases}$$

Note that equation (6.10) is equivalent to the following equality constraint

$$b_R = Fy,$$

where $y \in \{0,1\}^p$. Using this notation, we have that $F(\{0,1\}^p) = R^m$ and $F([0,1]^p) = [-(1-2^{-k})U, (1-2^{-k})U]^m$. Without loss of generality, we assume that $F([0,1]^p)$ contains $\mathrm{dom}(f)$. Thus, the optimal value function f can be represented as

$$\begin{aligned} f(b) = \min_{x,y,z} \quad & c^\top x + q^\top z \\ \text{s.t.} \quad & Fy = b, \\ & Ax - Fy + Gz \leq 0, \\ & (x,z) \in X, \ y \in [0,1]^p, \end{aligned} \tag{6.11}$$

see figure 6.8 for an illustration. We call this representation the binarization step.

The second step in direction of the Blessing of Binary is the lifting of the feature space of f. We define the *lifting* of f by the optimal value function ϕ obtained from (6.11) with $Fy = b$ replaced by $y = \lambda$:

$$\begin{aligned} \phi(\lambda) = \min_{x,y,z} \quad & c^\top x + q^\top z \\ \text{s.t.} \quad & y = \lambda \\ & Ax - Fy + Gz \leq 0, \\ & (x,z) \in X, \ y \in [0,1]^p. \end{aligned} \tag{6.12}$$

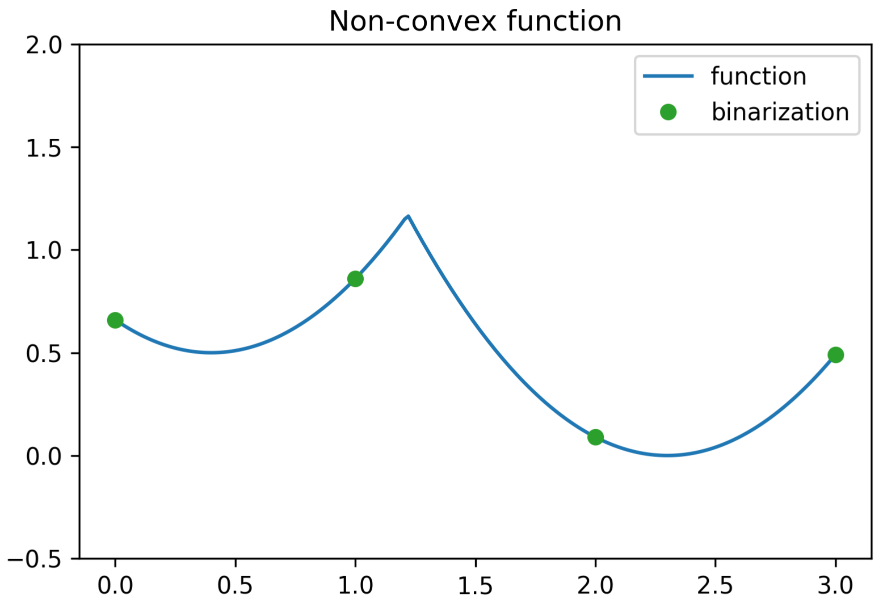
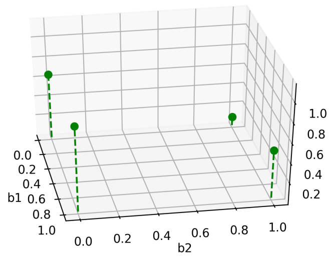


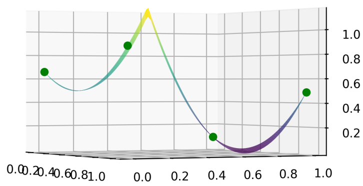
Figure 6.8: Binarization of a non-convex function

Note that the right-hand side dimension increases from m to $p = 2mk$, and we have that $\phi(\lambda) = f(F\lambda)$, for each $\lambda \in [0,1]^p$, see figures 6.9a, 6.9b and 6.9c. Recall that $F([0,1]^p)$ contains $\text{dom}(f)$, so the images of ϕ and f coincide. From the Blessing of Binary (Theorem 6.2), the biconjugate ϕ^{**} equals ϕ at the binary vectors $\hat{\lambda}$, see figures 6.9d and 6.9e. Regarding those ideas, the SDDiP algorithm obtain a binary state variable $\hat{\lambda} \in \{0,1\}^p$ in the forward step, and computes a Lagrangian cut for ϕ at $\hat{\lambda}$ in the backward step. Therefore, the resulting policy is an approximation of the optimal policy.

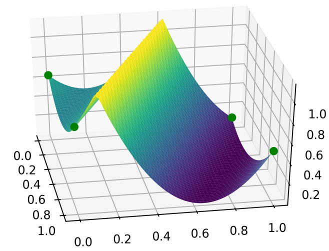
One relevant question is whether approximating a cost-to-go function f at a uniform grid lead to a near optimal policy. Given a stochastic MILP program, the paper of ZOU; AHMED; SUN [28] proves that we control the quality of such approximation by the uniform grid discretization error ϵ. The idea behind the proof relies on the fact that f is a Lipschitz function in $\text{dom}(f)$ (BLAIR; JEROSLOW [8]),



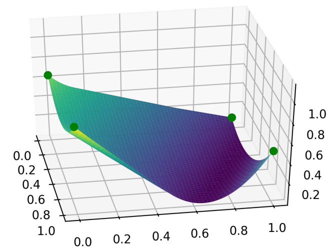
(a) State discretization.



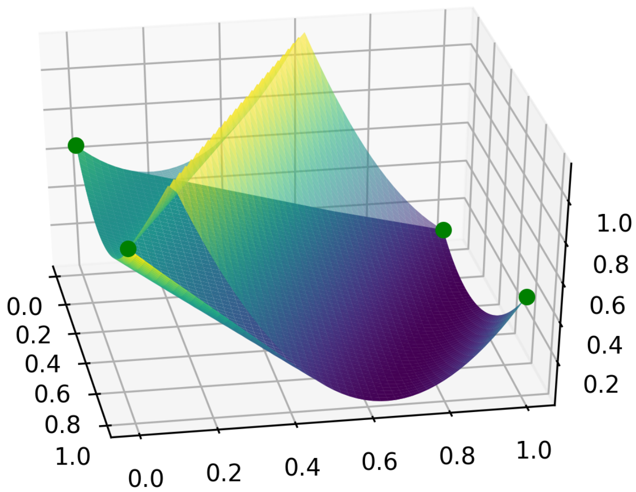
(b) Lift side view.



(c) Lift perspective.



(d) Lift convexification.



(e) Convexification gap.

Figure 6.9: SDDiP and the Blessing of Binary.

and for each $b \in \text{dom}(f)$, there is $\hat{\lambda}_R \in \{0,1\}^p$ such that $\|b - F\hat{\lambda}_R\| < \epsilon$. Therefore,

$$\|f(b) - \phi^{**}(\hat{\lambda}_R)\| = \|f(b) - f(b_R)\| \leq L\|b - b_R\| \leq L\epsilon.$$

Thus, the Lagrangian cuts for ϕ at the binary vectors $\hat{\lambda}_R \in \{0,1\}^p$ approximate f with arbitrary precision.

6.3 Case study: hydrothermal operational planning with Disjunctive Constraints

In this section, we present the results for the Disjunctive Constraints approach given by CAR-B and the Deficit Suppression Constraint (SPDef), so we compare it with the base case and the traditional penalization approach. These three modeling options are combined with two types of risk measures: the expected value and the mean-CVaR, which will be abbreviated by CVaR only.

Let us recall the base model. The objective function is to minimize the thermal and deficit costs $c_g^\top g_t + c_{df}^\top df_t$ plus the future cost-to-go function $\beta\overline{\mathcal{Q}}_{t+1}(v_{t+1})$:

$$\begin{aligned} Q_t(v_t, a_t) = \min_{x_t \in \mathcal{X}_t} \quad & c_g^\top g_t + c_{df}^\top df_t + \beta\overline{\mathcal{Q}}_{t+1}(v_{t+1}) \\ \text{s.t.} \quad & (*)_{\text{P}}, \end{aligned} \tag{6.13}$$

where x_t is the decision vector with all operational variables, $\mathcal{X}_t$ are the upper and lower bound limits of each variable, and $(*)_{\text{P}}$ represents the other operational constraints.

The CAR is a pre-established stored volume curve, denoted by v_{MinOP} (we omit the subscript t to not overload the notation). One possible use in the optimization model (6.13) considers an unit cost θ_t that penalizes the final stored volumes

v_{t+1} which lie below the CAR volume v_{MinOP}:

$$
\begin{aligned}
Q_t(v_t, a_t) = \quad \min_{x_t \in \mathcal{X}_t} \quad & c_g^\top g_t + c_{df}^\top df_t + \beta \overline{\mathcal{Q}}_{t+1}(v_{t+1}) + \theta_t^\top (v_{\text{MinOP}} - v_{t+1})_+ \\
\text{s.t.} \quad & (*)_{\text{P}}.
\end{aligned}
\tag{6.14}
$$

We call formulation (6.14) the Penalty (Pen) approach. For this example, we consider a constant vector v_{MinOP} equal to 20% of the maximum storage capacity $\overline{v}$, and a penalty value θ_t between the most expensive thermal cost and the cost of the first deficit level.

The CAR-B approach follows the same idea of CAR but uses binary variables, $z_g \in \{0, 1\}^{N_{\text{sys}}}$, to induce thermal generation if the final stored volume v_{t+1} is below the reference vector v_{MinOP}. We also consider the Suppression of Preventive Deficit (SPDef) constraint in this formulation that we call the Disjunctive Constraint (Disj.) approach:

$$
\begin{aligned}
Q_t(v_t, a_t) = \quad \min_{x_t \in \mathcal{X}_t} \quad & c_g^\top g_t + c_{df}^\top df_t + \beta \overline{\mathcal{Q}}_{t+1}(v_{t+1}) \\
\text{s.t.} \quad & (*)_{\text{P}}, \\
& (1 - z_g) \cdot v_{\text{MinOp}} \leq v_{t+1} \leq (1 - z_d) \cdot \overline{v}, \\
& M_I g_t \geq z_g \cdot \underline{G}, \qquad 0 \leq df_t \leq z_d \cdot d_t, \\
& z_g, z_d \in \{0, 1\}^{N_{\text{sys}}}.
\end{aligned}
\tag{6.15}
$$

We note that the security thermal dispatch is regional, that is, if the final stored volume $v_{i,t+1}$ of a subsystem i is below the reference value $v_{i,\text{MinOp}}$, then only the thermal plants from the subsystem i are dispatched. This is the reason for the indicator matrix M_I in the constraint $M_I g_t \geq z_g \underline{G}$. For this case study, the vector $\underline{G}$ has coordinates that equals the maximum thermal generation capacity of each subsystem, i.e., $\underline{G} = M_I \overline{g}$. The SPDef constraint is also regional, since it is only possible to have a load shedding in a subsystem i, $df_{i,t} > 0$, if the corresponding final stored volume $v_{i,t+1}$ is zero.

These three modeling proposals were combined with two coherent risk measures: the expected value and CVaR. Recall that a coherent risk measure ρ_t (SHAPIRO;

DENTCHEVA; RUSZCZYŃSKI [26]) is a function that associates random costs Z to real numbers and satisfies some axioms equivalent to the convexity definition. The risk measure appears in the definition of the future cost-to-go function $\overline{\mathcal{Q}}_{t+1}(v_{t+1})$:

$$\overline{\mathcal{Q}}_{t+1}(v_{t+1}) = \begin{cases} \rho_t \left[Q_{t+1}(v_{t+1}, a_{t+1}) \right] & , \quad t \in \{1, \ldots, T-1\}, \\ 0 & , \quad t = T, \end{cases} \tag{6.16}$$

where we call the function $\rho_t[Z] = \mathbb{E}[Z]$ the risk neutral case and $\rho_t[Z] = (1 - \lambda)\mathbb{E}[Z] + \lambda \mathsf{CVaR}_\alpha[Z]$ the CVaR case. The parameters used in the CVaR case are $\lambda = 0.10$ and $\alpha = 0.05$.

For the stochastic inflow a_t, we adopted a stagewise independent model with periodic empirical distribution on each month. The reason for this simplification is the difficulty in considering an autoregressive model along with the SDDiP algorithm, since the SDDiP requires upper and lower bound limits for the state variables. As the scenarios of an autoregressive model can assume very high values (and also very low values), we decided to simplify the inflow model.

In all cases, we consider four interconnected subsystems, $N_{\text{sys}} = 4$, based on the configuration of the Brazilian Interconnected Power System of January 2015, with a planning horizon of 5 years ($T = 60$ monthly stages) that ends in December 2019. Table 6.1 shows the maximum storable energy $\overline{v}$, the CAR reference storage v_{MinOp}, and the maximum thermal generation capacity $M_I \overline{g}$ of each subsystem. To better understand the effects of different policies (base case, Pen., Disj. ...), we increased the demand from each stage by a factor of 5%.

All the numerical experiments were performed on a computer with Intel(R) Core(TM) i7-4790, 8Gb RAM, operational system Ubuntu Xenial (16.04), and Julia language version 0.6 [6]. We used the packages `SDDP.jl` [14] and `SDDiP.jl` [19], and the resulting optimization problems, both linear and mixed integer linear programs were solved by Gurobi [16], version 7.0.2.

	Max. Storage $\overline{v}$	Storage Tgt. v_{MinOp}	Max. ThermGen. $M_I\overline{g}$
SE	204078.3	40815.7	10889
S	19929.2	3985.8	3085
NE	51806.1	10361.2	6152
N	15765.5	3153.1	319

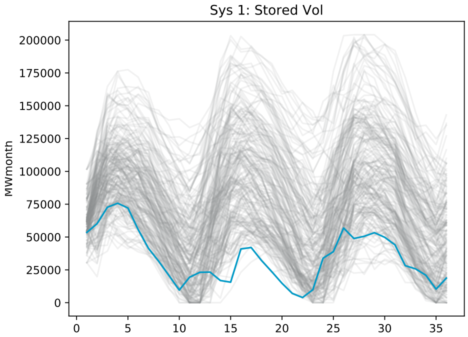
Table 6.1: Case study parameters in MWmed.

We present below the results obtained for each of the 6 models regarding the base case, Penalization and Disjunctive Constraints approach combined with the expected value and CVaR risk measures. After the 200 iterations of the SDDiP algorithm for each of these models, we performed an out-of-sample simulation considering 200 historical scenarios for the first 36 months instead of all the 60 months, in order to avoid the end of horizon effect in the policy estimation.

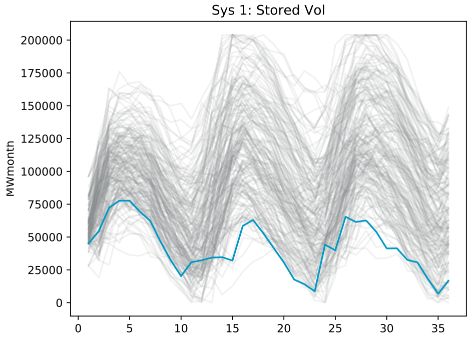
6.3.1 Stored energy

For the stored energy, we note in figure 6.10 that both traditional mechanisms, Penalization and CVaR were able to increase the stored volume along the stages.

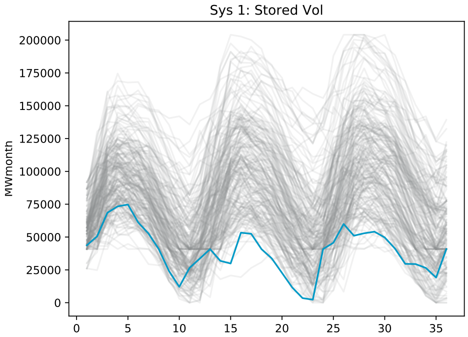
In comparison to the expected value, the CVaR risk measure increases the stored energy for the three policies in which it was used (base case, Disj., Pen.). The inclusion of the Disjunctive Constraints leads to an increase in the stored volume, mainly, during the dry season, as compared to the base case and for both risk measures. The Penalization approach (with and without CVaR) resulted in the lowest violation of the reference volume v_{MinOp}, as observed in figures 6.10e and 6.10f. However, as we shall see later, the Penalization approach also generates the greatest amount of deficit.



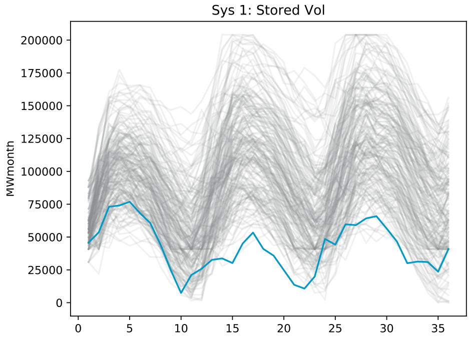
(a) Risk neutral.



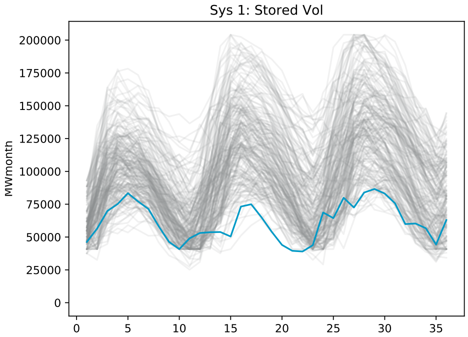
(b) CVaR.



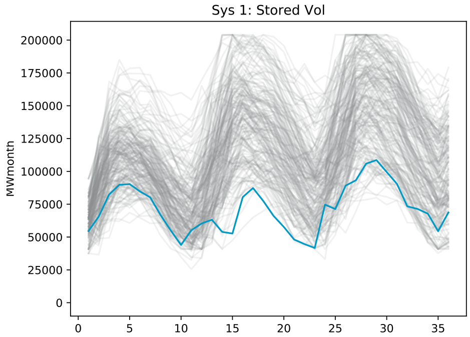
(c) Disj.



(d) CVaR+Disj.



(e) Pen.

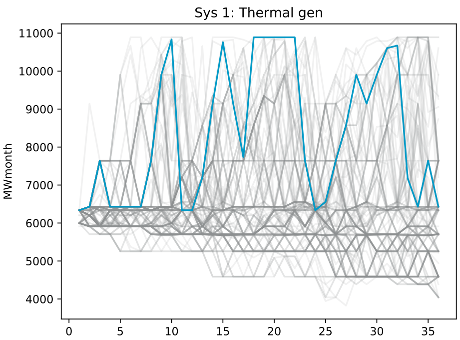


(f) CVaR+Pen.

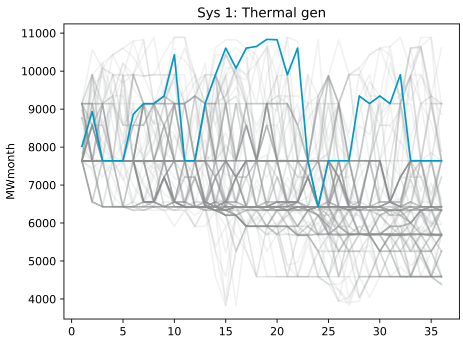
Figure 6.10: Stored energy.

6.3.2 Thermal generation

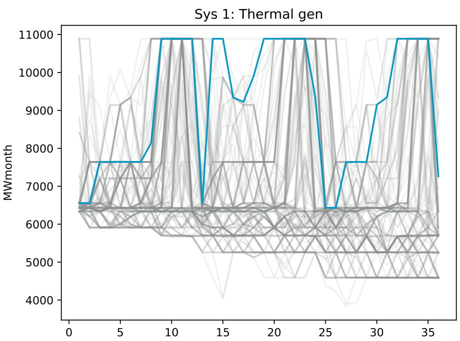
Figure 6.11 presents the 200 thermal generation scenarios for each policy, and again we clearly note the impact of CVaR. Combined or not with Penalization or Disjunctive Constraints, the CVaR increases the thermal dispatch, mainly at the early stages of the planning problem. We also note the effect of the Disjunctive Constraints proposal: it also increases the thermal dispatch, and for several scenarios the thermal generation stays in its full capacity. The inclusion of Penalization, see figure 6.11e, increases the thermal dispatch more sharply than Disjunctive Constraints for both risk measures, also in the early stages of the planning horizon. The combined use of Penalization and CVaR leads to the greatest thermal dispatch among all considered policies.



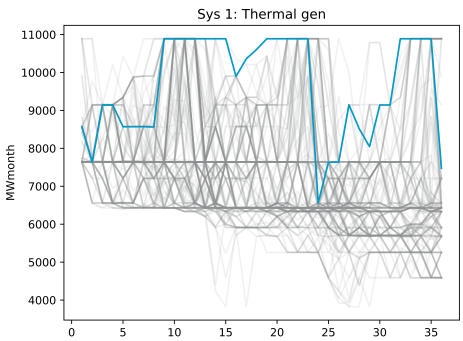
(a) Risk neutral.



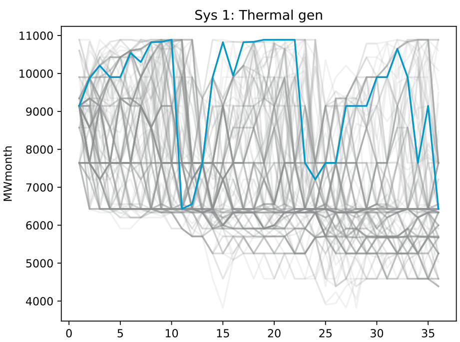
(b) CVaR.



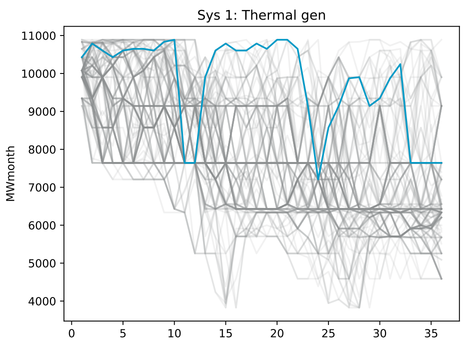
(c) Disj.



(d) CVaR+Disj.



(e) Pen.



(f) CVaR+Pen.

Figure 6.11: Thermal generation.

6.3.3 Deficit

The deficit (load shedding) is quite pronounced for the "pure" risk-neutral policy, with relatively frequent deficit scenarios with values up to 8.000 MWmonth, as shown in figure 6.12a. Both CVaR and Disjunctive Constraints alone reduce the occurrence of deficit, but their combined use leads essentially to its elimination (figure 6.12d).

However, the Penalization approach for the neutral risk case (figure 6.12e) leads to frequent occurrence of deficit, and even when combined with CVaR (figure 6.12f) does not significantly reduce this effect. On the other hand, the amplitude of the load shedding in the Disjunctive Constraints proposal when there is no CVaR is greater than that of the Penalization proposal, see figures 6.12c and 6.12e.

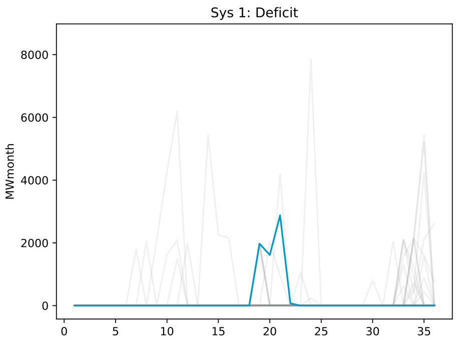
A more detailed analysis of the deficit variable is shown in table 6.2, where we discriminate the deficit variable by levels (depth of load shedding), and each table entry is the sum of the corresponding deficit occurrences along the 36 stages and 200 scenarios for each subsystem and each policy. A relevant effect of the Penalization policy is the significant increase of the total amount of deficit in the first level, both with respect to the risk neutral and CVaR case. The reason for this is that the penalty value θ_t turns the "water value" $\theta_t + |\frac{\partial \overline{Q}_{t+1}(v^*_{t+1})}{\partial v_{t+1}}|$ so significant that it becomes higher than the cost for the first deficit level, which leads to an optimal solution that induces deficit rather than using stored volume for power generation. Figure 6.13 illustrates the relative differences between the deficit values of the first level from table 6.2.

Def. Level		Risk Neutral 1	2	3	4	CVaR 1	2	3	4
Sub system	SE	89067	18383	8640	0	6853	0	0	0
	S	32231	5509	3878	0	6478	1276	0	0
	NE	910	455	463	0	459	0	0	0
	N	1152	336	569	0	285	0	0	0

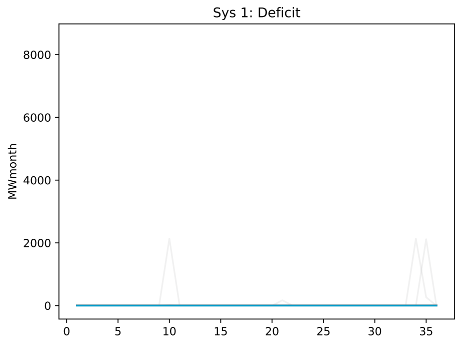
Def. Level		Pen. 1	2	3	4	CVaR + Pen. 1	2	3	4
Sub system	SE	219987	2136	0	0	98874	0	0	0
	S	69104	591	0	0	36252	0	0	0
	NE	413	0	0	0	1899	0	0	0
	N	885	0	0	0	1320	0	0	0

Def. Level		Disj. 1	2	3	4	CVaR + Disj. 1	2	3	4
Sub system	SE	13202	6096	3249	0	4	0	0	0
	S	5691	1837	270	0	372	0	0	0
	NE	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0

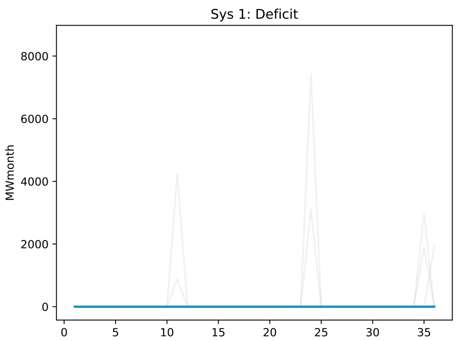
Table 6.2: Deficit average (MWmonth) along 36 stages and over 200 scenarios.



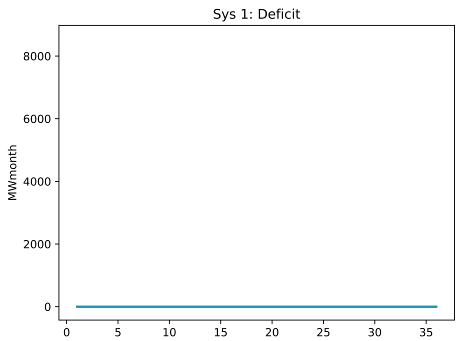
(a) Risk neutral.



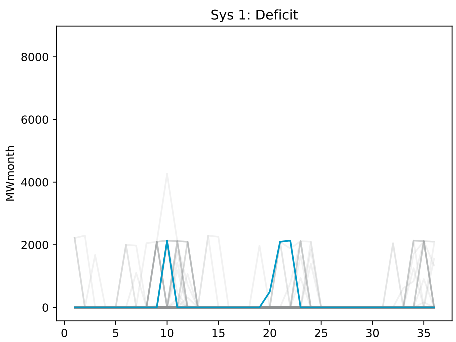
(b) CVaR.



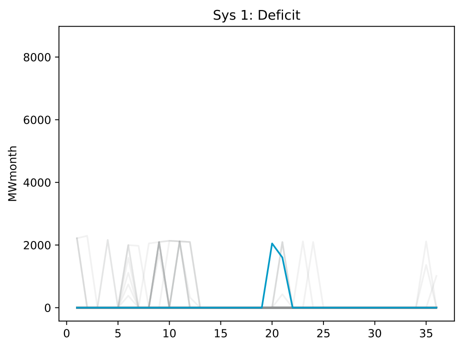
(c) Disj.



(d) CVaR+Disj.



(e) Pen.



(f) CVaR+Pen.

Figure 6.12: Load shedding.

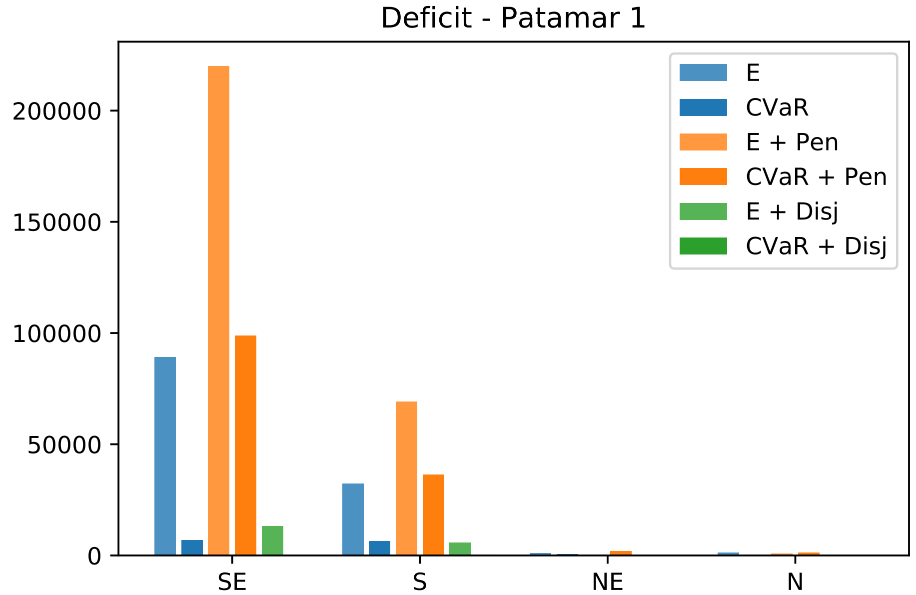


Figure 6.13: Deficit of level 1 (MWmonth) for each policy and subsystem.

6.3.4 Operational cost

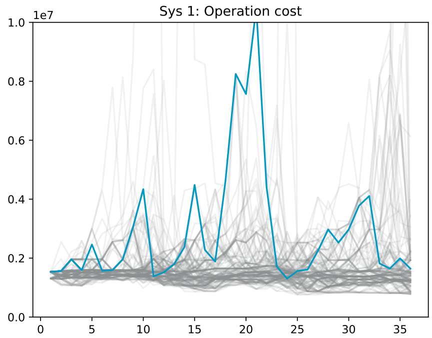
Figure 6.14 shows the operational cost series which comprises the total thermal cost plus the total deficit cost at each stage, but does not consider the penalties of the Penalization proposal, since they do not represent a real cost.

We note that the use of CVaR as a risk aversion approach has two effects: the average cost is higher when compared to the risk-neutral case (with or without Penalization or Disjunctive Constraints), and the variability of the operational cost is also lower. Regarding the Disjunctive Constraints approach, we do not see a clear increase in the average cost, but it is possible to notice that there was a reduction in the number of scenarios with extremely high costs (above 6 M$). This comment is valid for both the risk-neutral policy and the CVaR policy with Disjunctive Constraints, see figures 6.14c and 6.14d. Lastly, the Penalization proposal leads to an increase in operational costs, both on average and in the frequency of high costs.

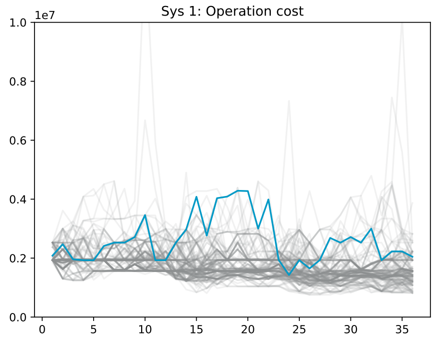
Table 6.3 shows the cost average of each policy over the first 36 months (stages) and 200 scenarios. We emphasize that the Penalization and CVaR penalties were not considered in the cost calculation. Note that CVaR alone results in a moderate cost increase (8.26 %), the combined use of Disjunctive Constraints and CVaR results in a even greater cost, but the Penalization approach with both risk neutral or CVaR lead to the greatest costs due to the occurrence of preventive deficit. Note that Penalization with CVaR produces an increase in the average operational cost that is nearly five times the increase produced by CVaR, and almost two times the increase caused by Disjunctive Constraints with CVaR.

Policy	E	Pen.	Disj.	CVaR	CVaR+Pen.	CVaR+Disj.
Cost (M$)	43 327	55 812	49 120	46 907	60 445	51 541
Increase (%)	–	28.82	13.37	8.26	39.51	21.85

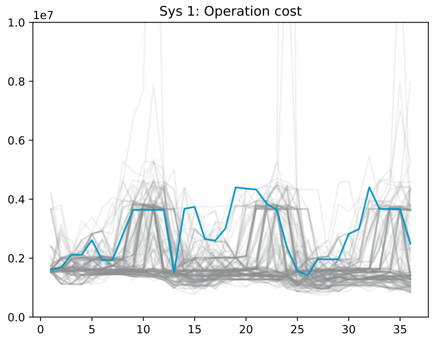
Table 6.3: Expected value of cost and relative increase of each policy.



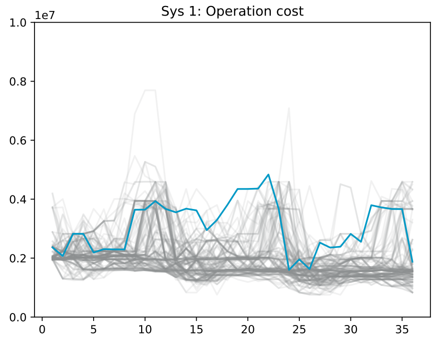
(a) Risk neutral.



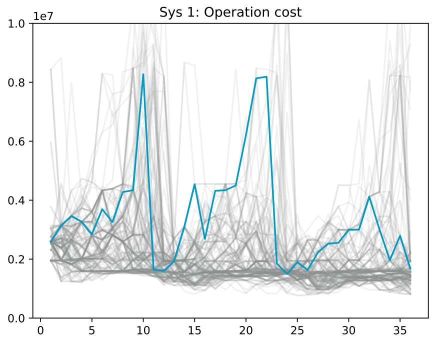
(b) CVaR.



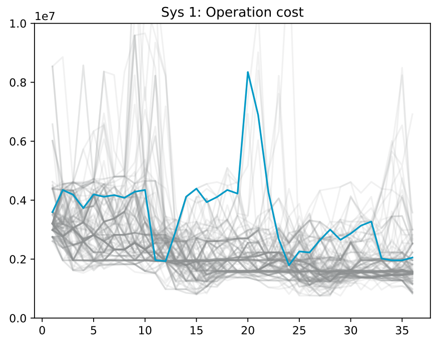
(c) Disj.



(d) CVaR+Disj.



(e) Pen.



(f) CVaR+Pen.

Figure 6.14: Operational cost.

6.3.5 Discretization error

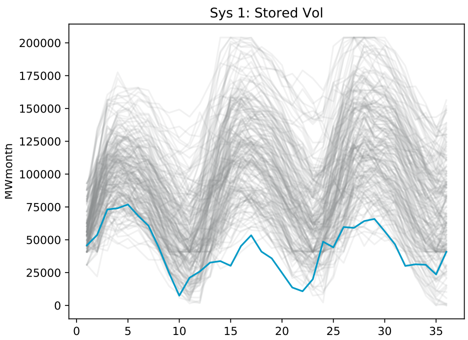
One of the important issues associated with the use of SDDiP is the need to discretize the state variables. In fact, the greater the precision of discretization, the closer we expect to be to the non-convex continuous model. However, this leads to an increase in computational time because it increases the size of the problem and the number of binary variables.

Figure 6.15 compares the policies for the CVaR and Disjunctive Constraints approach obtained by the SDDiP algorithm considering an accuracy of $\epsilon = 10$ MWmonth (figures on the left) and $\epsilon = 1000$ MWmonth (figures on the right) for the state variable: the stored energy v_t. Taking a discretization step 100 times greater corresponds to reducing the number of binary variables required in each subsystem by $7 \sim \log_2(100)$. We present the graphics of stored energy, hydro generation and thermal generation simulated for the same 200 scenarios.

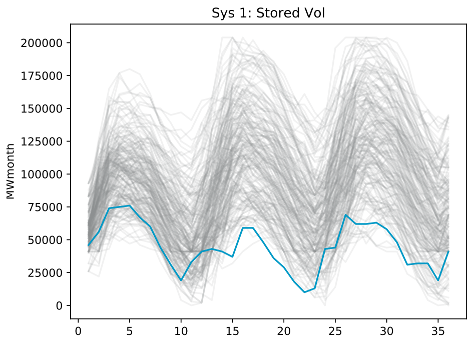
As shown in Figures 6.15a and 6.15b, there is no significant change in stored energy resulting from a larger discretization step. Also for figures 6.15c and 6.15d the hydro generation has a very similar overall behavior — although the result for the highlighted series is clearly different. However, we note a significant change regarding thermal generation: in the case of a discretization step of 10 MWmonth there are different "levels" of thermal generation which concentrate a large proportion of scenarios, while for the discretization error of 1000 MWmonth those levels are completely absent.

In order to understand why this happens, let us consider the typical behavior of these variables in the continuous case. Since the total thermal and deficit costs as a function of the final storage v_{t+1} is piecewise linear, there is a strong tendency for the corresponding optimal solution v^*_{t+1} to be in a 'kink' of this function. In other words, the optimal solution usually dispatches all the thermal plants at their full capacity up to a certain unit cost, and the hydro generation regulates the difference between the corresponding thermal generation and total demand.

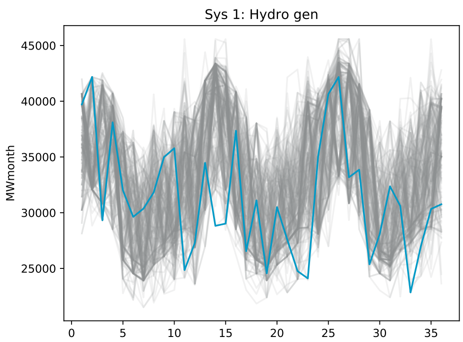
With the discretization of the stored volume, this mechanism is no longer free, and, unless for the spillage, the hydro generation is also “discretized” and able only to take values from 10 to 10 MWmonth or from 1000 em 1000 MWmonth. This reverses the “order” of constraints: first we determine the hydro generation and only then we obtain the thermal generation that meets the demand. Due to the variability of the inflow scenarios and depending on the discretization error, we do not obtain a thermal generation that follows on the same “levels” over some stages, as is more usual.



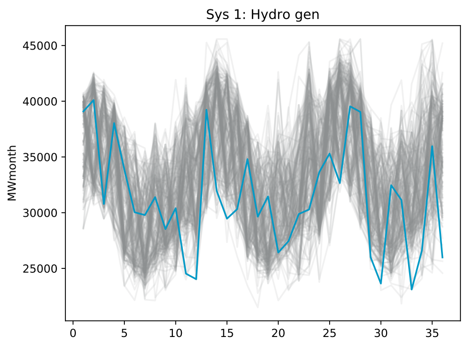
(a) Stored energy ($\epsilon = 10$)



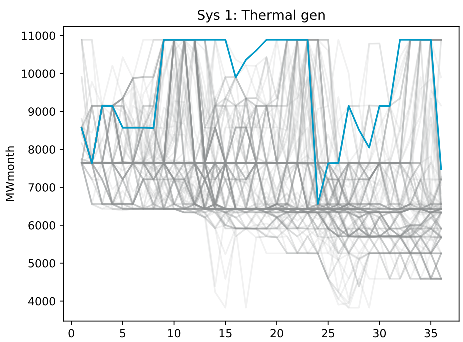
(b) Stored energy ($\epsilon = 1000$)



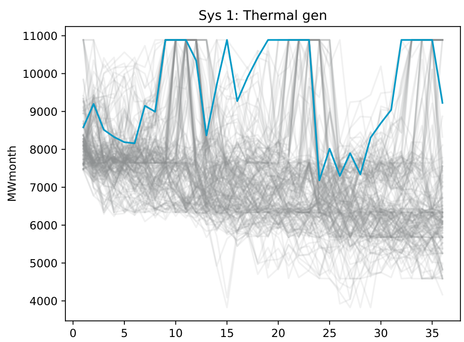
(c) Hydro generation ($\epsilon = 10$)



(d) Hydro generation ($\epsilon = 1000$)



(e) Thermal generation ($\epsilon = 10$)



(f) Thermal generation ($\epsilon = 1000$)

Figure 6.15: Impact of the grid precision ϵ of the state variable v_t (initial storage of stage t) in the policy estimation.

7 BLESSING OF EXTREME POINTS

In this chapter we revisit two results presented in this dissertation: Balas's theorem (Theorem 5.3 page 101) and the Blessing of Binary (Theorem 6.2 page 6.2). In particular, both theorems state that the convex hull operation does not add points to some special affine spaces. We prove that those affine spaces are closely related to particular extreme points, and we call that the "Blessing of extreme points". We provide some examples to illustrate this result.

7.1 Balas's theorem revisited

In this section, we investigate more closely Balas's theorem (Theorem 5.3 page 101), and how it can be interpreted geometrically. Let P_i be the polyhedron given by $\{x \in \mathbb{R}^n \mid A_i x \leq b_i\}$. Formula (5.1), given by

$$Q = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} x = \sum_{j=1}^p y_j,\ A_i y_i \leq z_i b_i,\ y_i \in \mathbb{R}^n, \\ \sum_{j=1}^p z_j = 1,\ z_i \in \{0,1\},\ i = 1, \ldots, p. \end{array} \right\},$$

represents the union of polyhedra $\bigcup_{i=1}^p (P_i + R)$, where $R = \sum_{i=1}^p \operatorname{recc}^A(P_i)$. Recall that $\operatorname{recc}^A(P_i) := \{d \in \mathbb{R}^n \mid A_i d \leq 0\}$ are the *algebraic* recession cones of the polyhedra P_i. From Theorem 5.3 (page 101), if we consider z_i as a continuous variable in $[0,1]$, the resulting set $\overline{Q}$ becomes the convex closure of Q, that is, $\overline{Q} = \operatorname{cl}\operatorname{conv} Q$. We view the set Q as the projection on the x variable of the following set:

$$Q_{\text{lift}} = \left\{ (x,z) \in \mathbb{R}^n \times \{0,1\}^p \;\middle|\; \begin{array}{c} x = \sum_{j=1}^p y_j,\ A_i y_i \leq z_i b_i,\ y_i \in \mathbb{R}^n, \\ \sum_{j=1}^p z_j = 1,\ i = 1, \ldots, p. \end{array} \right\}. \tag{7.1}$$

Note that Q_{lift} is the Cartesian product of each polyhedron $\widetilde{P}_i = P_i + R$ with an extreme point of the $(p-1)$-simplex:

$$Q_{\text{lift}} = \bigcup_{i=1}^{p} \left(\widetilde{P}_i \times \{e_i\} \right), \tag{7.2}$$

where $e_i \in \mathbb{R}^p$ is a canonical basis element. Let $\overline{Q}_{\text{lift}}$ be the set (7.1) with z in the continuous interval $[0,1]^p$:

$$\overline{Q}_{\text{lift}} = \left\{ (x,z) \in \mathbb{R}^n \times [0,1]^p \;\middle|\; \begin{array}{c} x = \sum_{j=1}^{p} y_j,\ A_i y_i \le z_i b_i,\ y_i \in \mathbb{R}^n, \\ \sum_{j=1}^{p} z_j = 1,\ i = 1,\dots,p. \end{array} \right\}. \tag{7.3}$$

The following theorem proves that $\overline{Q}_{\text{lift}}$ is the convex closure of Q_{lift}.

Theorem 7.1 (Lifted Balas)**.** *The convex closure of Q_{lift} equals $\overline{Q}_{lift}$, that is,*

$$\overline{Q}_{lift} = \operatorname{cl}\operatorname{conv}(Q_{lift}).$$

Proof. Let (x,z) be a vector of $\overline{Q}_{\text{lift}}$. For each i, there are $y_i \in \mathbb{R}^n$ such that

$$x = \textstyle\sum_{i=1}^{p} y_i, \quad A_i y_i \le z_i b_i, \quad \sum_{i=1}^{p} z_i = 1,$$

where $z_i \in [0,1]$, for all $i = 1,\dots,p$. Let I^0 and I^+ be the set of indexes i such that z_i is equal to zero and greater than zero, respectively. Note that $d_i = y_i$ belongs to $\operatorname{recc}(P_i)$, for all $i \in I^0$, and $\widetilde{x}_i = y_i/z_i$ belongs to P_i, for all $i \in I^+$. So, the vector x has the following form:

$$x = \sum_{i\in I^+} z_i \frac{y_i}{z_i} + \sum_{i \in I^0} d_i = \sum_{i \in I^+} z_i \widetilde{x}_i + \sum_{i\in I^0} d_i,$$

and z can also be represented as

$$z = \sum_{i=1}^{p} z_i e_i = \sum_{i \in I^+}^{p} z_i e_i + \sum_{i \in I^0}^{p} 0 e_i.$$

So, the pair (x,z) can be represented by the following sum:

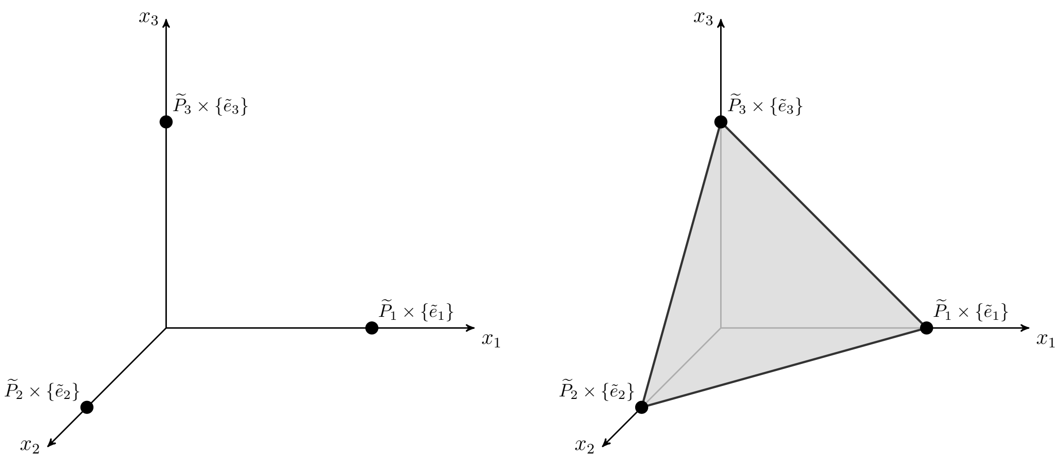
$$(x,z) = \underbrace{\sum_{i\in I^+} z_i \left(\widetilde{x}_i, e_i\right)}_{\operatorname{conv}(\bigcup_{i=1}^{p} \widetilde{P}_i \times \{e_i\})} + \underbrace{\sum_{i \in I^0} (d_i, 0)}_{\sum_{i=1}^{p} \operatorname{recc}(\widetilde{P}_i)\times\{0\}}.$$

From Theorem 2.13 (page 33) and using (7.2), the convex closure of Q_{lift} has also the following expression:

$$\operatorname{cl}\operatorname{conv}(Q_{\text{lift}}) = \operatorname{conv}\left(\bigcup_{i=1}^{p}(\widetilde{P}_i \times \{e_i\})\right) + \sum_{i=1}^{p} \operatorname{recc}(\widetilde{P}_i \times \{e_i\}),$$

Since $\operatorname{recc}(P_i \times \{e_i\}) = \operatorname{recc}(P_i) \times \{0\}$, we have that (x, z) belongs to $\operatorname{cl}\operatorname{conv}(Q_{\text{lift}})$. Therefore, $\overline{Q}_{\text{lift}}$ is a subset of $\operatorname{cl}\operatorname{conv}(Q_{\text{lift}})$.

On the other hand, the set $\overline{Q}_{\text{lift}}$ is closed, convex and contains Q_{lift}. Thus, we have that $\overline{Q}_{\text{lift}}$ contains $\operatorname{cl}\operatorname{conv}(Q_{\text{lift}})$, which concludes the result. □



(a) Lifted Balas set Q_{lift} (pictorial) . (b) Lifted Balas convex closure $\overline{Q}_{\text{lift}}$.

Figure 7.1: Pictorial representation of the Lifted Balas theorem.

Balas's theorem (Theorem 5.3 page 101) is a corollary of the Lifted Balas theorem, and the explanation of this relationship leads to interesting geometric insights. Before proceeding to the proof, we need to establish some properties regarding the commutation between linear maps and the convex hull, $\operatorname{conv}(\cdot)$, and convex closure, $\operatorname{cl}\operatorname{conv}(\cdot)$, operators.

Lemma 7.2 (Convexness of image under linear transformation)**.** *Let $A \in \mathbb{R}^{m \times n}$ be a linear map, and let D be any subset of $\mathbb{R}^n$. Then, the convex hull operator* *commutes with A:*

$$A\,\mathrm{conv}(D) = \mathrm{conv}(AD).$$

In particular, if D is convex then A is also convex.

Proof. This is a straightforward argument of commuting convex combinations and linear maps. Let $y \in A\,\mathrm{conv}(D)$. There exist $x_i \in D$ and $\lambda_i \in [0,1]$ for all $i = 1,\dots,k$ such that $\sum_{i=1}^{k} \lambda_i = 1$, and

$$y = A\left(\sum_{i=1}^{k} \lambda_i x_i\right) = \sum_{i=1}^{k} \lambda_i A x_i \in \mathrm{conv}(AD).$$

Let $y \in \mathrm{conv}(AD)$. There exist $x_i \in D$ and $\lambda_i \in [0,1]$ for all $i = 1,\dots,k$ such that $\sum_{i=1}^{k} \lambda_i = 1$, and

$$y = \sum_{i=1}^{k} \lambda_i A x_i = A\left(\sum_{i=1}^{k} \lambda_i x_i\right) \in A\,\mathrm{conv}(D).$$

If D is convex, then $\mathrm{conv}(D) = D$. So, $\mathrm{conv}(AD) = AD$ which is equivalent to AD being convex. $\square$

The following theorem is the core argument for proving the Balas theorem from the Lifted Balas result. It states that any linear map commutes with the convex closure operator for any finite union of polyhedra. This result can also be extended to general closed convex sets with some regularity conditions, but its proof is unnecessary for this section.

Lemma 7.3 (Commutation of the convex closure operator with a linear map)**.** *Let A be a linear map, and let P_i be polyhedra, for $i = 1,\dots,p$. Then,*

$$A\left[\mathrm{cl}\,\mathrm{conv}\left(\bigcup_{i=1}^{p} P_i\right)\right] = \mathrm{cl}\,\mathrm{conv}\left[\bigcup_{i=1}^{p} AP_i\right].$$

Proof. This lemma results from the following relations:

$$\begin{aligned} A\left[\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{p} P_i\right)\right] &= A \operatorname{conv}\left(\bigcup_{i=1}^{p} P_i\right) + \sum_{i=1}^{p} A \operatorname{recc}(P_i). \\ &= \operatorname{conv}\left[A\left(\bigcup_{i=1}^{p} P_i\right)\right] + \sum_{i=1}^{p} \operatorname{recc}(AP_i) \\ &= \operatorname{conv}\left(\bigcup_{i=1}^{p} AP_i\right) + \sum_{i=1}^{p} \operatorname{recc}(AP_i) \\ &= \left[\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{p} AP_i\right)\right], \end{aligned}$$

where the first equality follows from Theorem 2.13 (page 33), the second equality follows from Lemma 7.2 just proved and Theorem 2.8 (page 25), the third equality is noting that $A\,(\bigcup_{i=1}^{p} P_i) = \bigcup_{i=1}^{p} AP_i$, and the fourth equality follows again from Theorem 2.13 (page 33). ☐

In the particular case where A is a projection, Lemma 7.3 yields

Theorem 7.4 (Balas revisited)**.** *The convex closure of Q equals $\overline{Q}$, that is,*

$$\overline{Q} = \operatorname{cl}\operatorname{conv} Q.$$

Proof. Recall that Q and $\overline{Q}$ are the projection in x variable of Q_{lift} and $\overline{Q}_{\text{lift}}$, respectively. This proof results from the following equations:

$$\begin{aligned} \overline{Q} &= \operatorname{proj}_x \overline{Q}_{\text{lift}} \\ &= \operatorname{proj}_x \left[\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{p} (\widetilde{P}_i \times \{e_i\})\right)\right] \\ &= \operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{p} \widetilde{P}_i\right) = \operatorname{cl}\operatorname{conv} Q, \end{aligned}$$

where the first equality follows from the definition of $\overline{Q}$ and $\overline{Q}_{\text{lift}}$, the second equality is the Lifted Balas theorem, the third equality follows from Lemma 7.3 taking $A = proj_x(\cdot)$, and the fourth equality follows from the geometric description of Q. $\square$

We note that if any "magic" formula provides the description of the convex closure in a higher dimensional space, we can project it to have the convex closure in the original space.

A remarkable feature of the Lifted Balas set Q_{lift} is that when the variable z equals a simplex vertex e_i the resulting set becomes $\widetilde{P}_i \times e_i$:

$$Q_{\text{lift}} \cap (\mathbb{R}^n \times \{e_i\}) = \widetilde{P}_i \times \{e_i\},$$

see formula 7.1. Even more surprisingly, the convex closure $\overline{Q}_{\text{lift}}$ has also the same property, that is,

$$\overline{Q}_{\text{lift}} \cap (\mathbb{R}^n \times \{e_i\}) = \widetilde{P}_i \times \{e_i\},$$

as we can see in formula 7.3. Actually, this is a general property of sets of the form $\bigcup_{i=1}^p (P_i \times \{r_i\})$. We call those sets the lift of $\bigcup_{i=1}^p P_i$ along $\{r_1, r_2, \ldots, r_p\}$. If $r_1, \ldots, r_p$ are extreme points from a convex set, Theorem 7.5 below proves that the convex hull does not add points to the affine space $\mathbb{R}^n \times \{\tilde{r}_i\}$, that is,

$$\text{conv}\left(\bigcup_{i=1}^{p} (P_i \times \{r_i\})\right) \cap (\mathbb{R}^n \times \{e_i\}) = P_i \times \{e_i\}. \tag{7.4}$$

This is the discrete version of the *Blessing of extreme points*. Actually, equation (7.4) also holds for the convex closure of $\bigcup_{i=1}^p (P_i \times \{r_i\})$, since the closure of intersection is the intersection of closures, because the intersection between the relative interiors of $P_i \times \{r_i\}$ and $\mathbb{R}^n \times \{e_i\}$ is nonempty, see BERTSEKAS [5] page 32 or ROCKAFELLAR [22] page 47. We provide a pictorial illustration of an extremal lift in figure 7.3.

Theorem 7.5 (Blessing of extreme points – discrete version)**.** *Let $D_1, D_2, \ldots, D_p$ be convex sets and $r_1, r_2, \ldots, r_p$ be extreme points of a convex set. Then, the convex hull of the finite union of Cartesian products $D_i \times \{r_i\}$ preserves its generators, that is,*

$$\operatorname{conv}\left(\bigcup_{i=1}^{p}(D_i \times \{r_i\})\right) \cap (\mathbb{R}^n \times \{r_j\}) = D_j \times \{r_j\}, \quad \textit{for all } j.$$

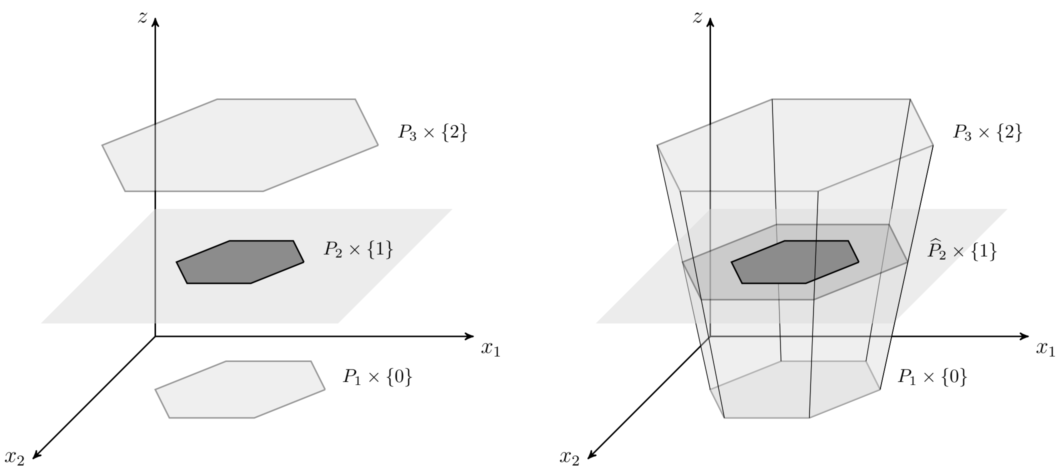
Proof. Let us denote by V the set $\operatorname{conv}(\cup_{i=1}^{p}(D_i \times \{r_i\}))$. By definition, $D_j \times \{r_j\}$ is a subset of V. So, $D_j \times \{r_j\}$ is a subset of $V \cap (\mathbb{R}^n \times \{r_j\})$, for each j.

To prove the reverse inclusion, let $x = (u, r_j)$ be an element of $V \cap (\mathbb{R}^n \times \{r_j\})$. Then, there are $(x_i, r_i) \in D_i \times \{r_i\}$, and scalars $\lambda_i \geq 0$ such that $\sum_{i=1}^{p} \lambda_i = 1$ and

$$(u, r_j) = \sum_{i=1}^{p} \lambda_i (x_i, r_i).$$

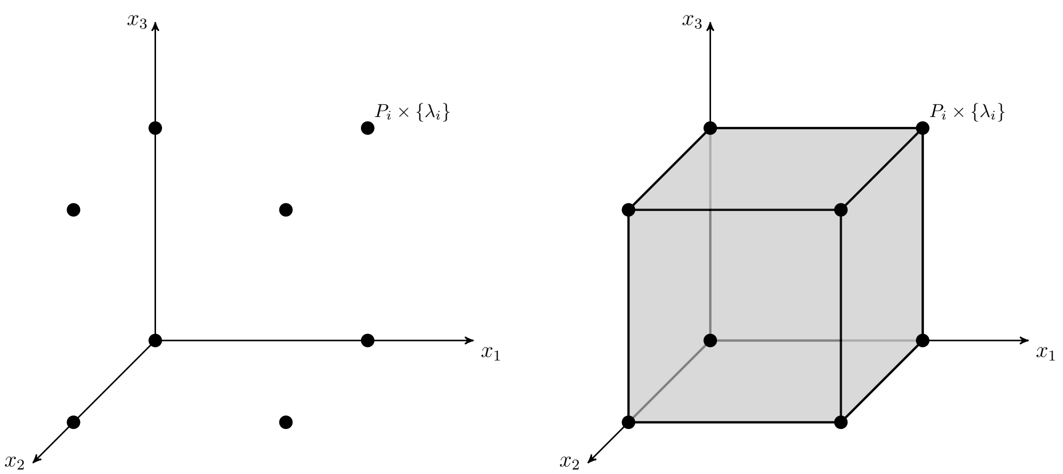
In particular, r_j is a convex combination of $r_1, \ldots, r_p$. Since, r_j is an extreme point, we have that $\lambda_j = 1$ and $\lambda_i = 0$, for all $i \neq j$. Therefore, the vector (u, r_j) equals (x_j, r_j), so x belongs to $D_j \times \{r_j\}$. $\square$

Figure 7.2, is a counterexample of Theorem 7.5 for a non-extremal lift. Figure 7.3 provides a pictorial illustration for the Blessing of extreme points using the vertices of a cube, which recall the binarization of state variables used in SDDiP. The Blessing of extreme points is a geometric feature that both the Balas theorem and the Blessing of Binary (Theorem 6.2, page 121) have in common, as we show in next section.



(a) Non-extremal lift. (b) Loss of information.

Figure 7.2: Non-extremal lifted set and the corresponding convex closure.



(a) Extremal lift. (b) Preservation of information.

Figure 7.3: The Blessing of extreme points using the vertex of a cube.

7.2 Blessing of Binary revisited

The Blessing of Binary (Theorem 6.2, page 121) states that, for a particular class of 0-1 MILP optimal value functions ϕ, the convex regularization ϕ^{**} equals the original function ϕ at the binary vectors $\lambda^* \in \{0,1\}^p$:

$$\phi^{**}(\lambda^*) = \phi(\lambda^*), \quad \text{for all } \lambda^* \in \{0,1\}^p.$$

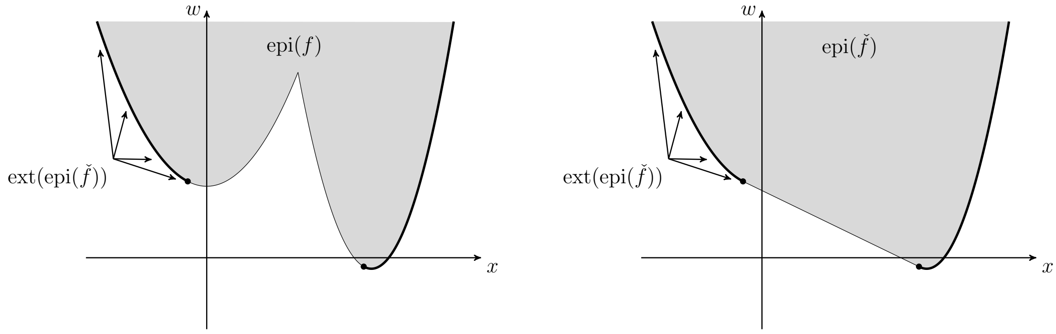
In this section, we complete the link between, the Blessing of Binary theorem and Balas's theorem from a geometrical perspective. We consider piecewise proper closed convex functions f, i.e., functions f that are the minimum of a finite number of proper closed convex function $f_1, \ldots, f_p$, and we conclude that $f^{**}(x)$ equals $f(x)$ for every extreme points x of $\text{dom}(f^{**})$. This result includes the Blessing of Binary, since ϕ is a piecewise polyhedral whose essential domain is contained in the unit cube $[0,1]^p$.

Recall the notation $\check{f}$ for the convex regularization of f, and its basic equation

$$\text{epi}(\check{f}) := \text{cl}\,\text{conv}(\text{epi}(f)),$$

and that, in most cases, $\check{f}$ equals the biconjugate f^{**}, see section 3.4. We adopt the notation $\check{f}$ instead of f^{**} to emphasize the geometrical properties of the convex regularization rather than the algebraic ones. We shall prove (Corollary 7.7) that if f is the minimum of a finite number of proper closed convex functions and $x^* \in \mathbb{R}^n$ is such that $(x^*, \check{f}(x^*))$ is an extreme point of $\text{epi}(\check{f})$, then there is no gap between $\check{f}$ and the original function f at x^*. The idea behind the proof is that given a finite union of nonempty closed convex sets $\bigcup_{i=1}^p D_i$, every extreme point of the convex closure $\text{cl}\,\text{conv}(\bigcup_{i=1}^p D_i)$ is also an extreme point of some convex set D_i. This property is the continuous version of the *Blessing of extreme points*.

Theorem 7.6 (Blessing of extreme points – continuous version)**.** *Let* $D_1, D_2, \ldots, D_p$ *be nonempty closed convex sets. Then, the set of extreme points of* $\text{cl}\,\text{conv}(\cup_{i=1}^p D_i)$



(a) Extreme points of epi($\check{f}$) in epi(f). (b) Extreme points of epi($\check{f}$).

Figure 7.4: Illustration of the Blessing of extreme points – continuous version and the Corollary 7.7.

is contained in the union of extreme points of D_i's:

$$\operatorname{ext}\left(\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{p} D_i\right)\right) \subseteq \bigcup_{i=1}^{p} \operatorname{ext}(D_i). \tag{7.5}$$

Proof. By Corollary 2.12 (page 31), the following equation holds

$$\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{p} D_i\right) = \operatorname{conv}\left(\bigcup_{i=1}^{p} D_i\right) + \sum_{i=1}^{p} \operatorname{recc}(D_i),$$

or the set $\operatorname{cl}\operatorname{conv}\left(\cup_{i=1}^{p} D_i\right)$ does not have any extreme point. In this second case, the relationship (7.5) is trivial. Suppose that the set of extreme points of $\operatorname{cl}\operatorname{conv}\left(\cup_{i=1}^{p} D_i\right)$ is nonempty, and let x^* be an extreme point. Thus, there are $y \in \operatorname{conv}\left(\cup_{i=1}^{p} D_i\right)$ and $d \in \sum_{i=1}^{p} \operatorname{recc}(D_i)$ such that

$$x^* = y + d.$$

Note that d must be the null vector, otherwise x^* could be expressed by a non-trivial convex combination of y and $y + 2d$, where both belongs to $\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{p} D_i\right)$:

$$x^* = \frac{1}{2}y + \frac{1}{2}(y + 2d).$$

Therefore, $x^* \in \operatorname{conv}\left(\cup_{i=1}^{p} D_i\right)$. Note that x^* belongs to D_j, for some j, otherwise x^* would be a non-trivial convex combination of vectors from $\cup_{i=1}^{p} D_i$. In particular, the element x^* is an extreme point of D_j. □

Corollary 7.7. *Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be the minimum of a finite number of proper closed convex functions. Then,*

$$\check{f}(x^*) = f(x^*),$$

for all $x^* \in \mathbb{R}^n$ such that $(x^*, \check{f}(x^*))$ is an extreme point of $\operatorname{epi}(\check{f})$.

Proof. Just apply Theorem 7.6 to $D_i = \operatorname{epi}(f_i)$. □

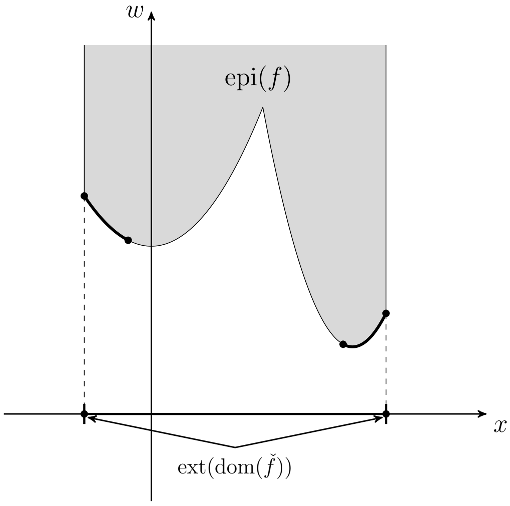
Actually, it may be difficult to check in practice whether the pair $(x^*, \check{f}(x^*))$ is an extreme point of $\operatorname{epi}(\check{f})$, since, in general, we do not even know $\operatorname{epi}(\check{f})$ explicitly. Fortunately, Corollary 7.8 states that if $x^* \in \mathbb{R}^n$ is an extreme point of $\operatorname{dom}(\check{f})$, then the pair $(x^*, \check{f}(x^*))$ is an extreme point of $\operatorname{epi}(\check{f})$, see figure 7.5. The Blessing of binary (Theorem 6.2, page 121) is the equivalent to Corollary 7.8 applied to a particular optimal value function f whose essential domain $\operatorname{dom}(f)$ is a high-dimensional cube.

Corollary 7.8. *Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be the minimum of a finite number of proper closed convex function. If x^* is an extreme point of* $\operatorname{dom}(\check{f})$*, then $(x^*, \check{f}(x^*))$ is an extreme point of* $\operatorname{epi}(\check{f})$*. In particular, $\check{f}(x^*) = f(x^*)$.*

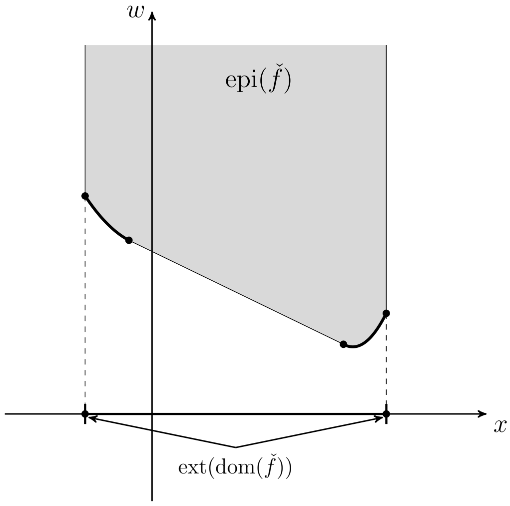
Proof. Let x^* be an extreme point of $\operatorname{dom}(\check{f})$. Since $\operatorname{dom}(\check{f})$ is a projection of $\operatorname{epi}(\check{f})$ on x^*, there is $w \in \mathbb{R}$ such that $(x^*, w) \in \operatorname{epi}(\check{f})$. Note that

$$\check{f}(x^*) = \inf_{(x^*, w) \in \operatorname{epi}(\check{f})} w.$$

Suppose that $(x^*, \check{f}(x^*))$ is not an extreme point of $\operatorname{epi}(\check{f})$, and let (x_1, w_1) and (x_2, w_2) be two vectors of $\operatorname{epi}(\check{f})$ that describe $(x^*, \check{f}(x^*))$ by a nontrivial convex combination. Then, $x_1 = x_2 = x^*$, since x^* is an extreme point of $\operatorname{dom}(\check{f})$. Moreover, at least one scalar w_1 or w_2 is strictly greater then $\check{f}(x^*)$, so a non trivial convex produces an scalar $\tilde{w}$ that is also strictly greater than $\check{f}(x^*)$. Therefore, the vector $(x^*, \check{f}(x^*))$ is an extreme point of $\mathrm{epi}(\check{f})$, so we conclude that $\check{f}(x^*) = f(x^*)$ by Corollary 7.7. $\square$



(a) Extreme points $x^* \in \mathrm{dom}(\check{f})$.



(b) Zero gap: $\check{f}(x^*) = f(x^*)$.

Figure 7.5: Extreme points of dom(f) have zero regularization gap.

8 CONCLUSION

This dissertation deals with stochastic optimization considering mixed integer linear constraints, which constitutes an appropriate framework to model real life applications. We provided two important results regarding the modeling of non-convex constraints with binary variables: a geometric interpretation of the Disjunctive Constraints formula (eq. (5.1), page 97), and a simpler statement and proof of the original Jeroslow's theorem about representability of union of polyhedra (JEROSLOW [18]). The name for the Jeroslow's theorem could be "Fundamental Theorem of Disjunctive Programming", since it asserts that the Disjunctive Constraint formula (5.1) represents the corresponding union of polyhedra if and only if that union is representable by a set of linear constraints involving continuous and binary variables. The Disjunctive Constraints formula (5.1) can be used in several real life applications, such as modeling operative constraints related to the power system planning operation. One instructive example is the hydropower plant minimum outflow constraint, which is a non-convex feasible set that would be hard to model without this theory.

Using the Disjunctive Constraints theory, we proposed a binary model for the risk aversion of low stored volumes, which we call the "disjunctive constraint approach", in such a way that if the stored volume falls below a reference value in a given subsystem then all the thermals plants from that subsystem are dispatched to their maximum capacity. We compared this approach with the standard ones such as the CVaR risk measure and the penalty model induced by CAR (Risk Aversion Curve). We observed that the penalty approach produces a significant amount of preventive deficit when the stored volume reaches the storage reference curve. The CVaR approach has interesting results in term of increasing the stored volume along the stages, and that happens because the CVaR risk measure penalizes high operational costs, and low stored volumes are indirectly related with high operational costs. The disjunctive constraint approach resulted in a better performance than the CAR penalty approach, since the disjunctive constraints do not induce preventive deficit and also dispatches thermal generation in low storage situations. Among all cases, the best one was the combination between the disjunctive constraints and the CVaR risk measure, since that combination provided high stored volumes and the least amount of deficit. Thus, the combination of CVaR and the disjunctive constraints seems a robust methodology for considering precisely both the operational rules and the risk averseness of low stored volumes. Thanks to the SDDiP algorithm of ZOU; AHMED; SUN [28], we were able to compare the standard approaches with the disjunctive approach, which consists in a multistage stochastic mixed integer linear program.

The SDDiP algorithm is supported by an important theorem called "Blessing of Binary" (BOB) (ZOU; AHMED; SUN [28]), which states that if f belongs to a class of MILP value functions, then the convex regularization $\check{f}$ equals the non-convex function f at the binary coordinates of the unit cube, where that unit cube contains the domain of $\check{f}$. In chapter 7, we proved the "Blessing of Extreme Points" (BEP), a result that extends and gives a geometrical interpretation for BOB based on the extreme points of the convex closure of union of closed convex sets $\overline{D} = \operatorname{cl}\operatorname{conv}(\bigcup_{i=1}^{p} D_i)$. The BEP theorem states that every extreme point of $\overline{D}$ is also an extreme point of some set D_k:

$$\operatorname{ext}(\overline{D}) \subset \bigcup_{i=1}^{p} \operatorname{ext}(D_i).$$

Using BEP, we have shown that if f is a piecewise convex function, then there is no gap between the convex regularization $\check{f}$ and the original function f at the extreme points of the domain of $\check{f}$. This expands the BOB's theorem to a larger class of functions.

As a final remark, we emphasize the importance of the interpretation of functions in terms of sets, and operations on functions in terms of operations on sets. Due to this geometric view, we were able to describe in which points the Lagrangian Relaxation of pointwise minima of convex functions have zero duality gap. We have shown that such a nonconvex function can be represented by a union of convex sets, $\cup_{i=1}^{p} D_i$, the corresponding Lagrangian Relaxation can be equivalently described by the convex closure of this union, $\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{p} D_i\right)$, and the points which are preserved under the convex closure operation are the extreme points of $\overline{D}$. We called this property the Blessing of Extreme Points (BEP) property. It is worth mentioning that the BEP proof was based on the following formula for the convex closure of union of closed convex sets:

$$\operatorname{cl}\operatorname{conv}\left(\bigcup_{i=1}^{p} D_i\right) = \operatorname{conv}\left(\bigcup_{i=1}^{p} D_i\right) + \sum_{i=1}^{p} \operatorname{recc}(D_i). \tag{8.1}$$

At first, we have found a proof of (8.1) on a paper of CERIA; SOARES [12]. However, there was a mistake in that paper (page 601), since the authors claim that equation (8.1) holds for any closed convex sets based on the assumption that projection of closed convex sets is always a closed. In chapter 2, we have shown a counter-example for both claims, and we dedicated a whole chapter for the proof of equation (8.1) under mild regularity conditions for the closed convex sets D_i's.

REFERENCES

[1] BALAS, E. Disjunctive Programming. In: **Annals of Discrete Mathematics**. [S.l.]: Elsevier, 1979. p.3–51.

[2] BALAS, E. Disjunctive programming: properties of the convex hull of feasible points. **Discrete Applied Mathematics**, [S.l.], v.89, n.1-3, p.3–44, dec 1998.

[3] BALAS, E.; CERIA, S.; CORNUÉJOLS, G. A lift-and-project cutting plane algorithm for mixed 0–1 programs. **Mathematical Programming**, [S.l.], v.58, n.1-3, p.295–324, jan 1993.

[4] BENDERS, J. F. Partitioning procedures for solving mixed-variables programming problems. **Numerische mathematik**, [S.l.], v.4, n.1, p.238–252, 1962.

[5] BERTSEKAS, D. P. **Convex Optimization Theory**. 1st.ed. [S.l.]: Athena Scientific, 2009.

[6] BEZANSON, J. et al. Julia: a fresh approach to numerical computing. **SIAM Review**, [S.l.], v.59, n.1, p.65–98, 2017.

[7] BIRGE, J. R.; LOUVEAUX, F. **Introduction to Stochastic Programming (Springer Series in Operations Research and Financial Engineering)**. [S.l.]: Springer, 2011.

[8] BLAIR, C.; JEROSLOW, R. The value function of a mixed integer program: i. **Discrete Mathematics**, [S.l.], v.19, n.2, p.121–138, 1977.

[9] CAMPELLO, R. E. **Cortes Disjuntivos para o problema do particionamento**. 1980. Tese (Doutorado em Ciência da Computação) — Federal University of Rio de Janeiro/COPPE.

[10] CAMPELLO, R. E.; FILHO, N. M. Aspectos Teóricos e Computacionais dos Cortes Disjuntivos B(4) e B(5). In: **Anais do XII Simpósio Brasileiro de Pesquisa Operacional - SOBRAPO**. [S.l.: s.n.], 1980.

[11] CAMPELLO, R. E.; FILHO, N. M. Algoritmo Dual de Cortes para o Problema de Programaçao Quadrática Não-Convexo. In: **XVI Simpósio Brasileiro de Pesquisa Operacional - SOBRAPO**. [S.l.: s.n.], 1981.

[12] CERIA, S.; SOARES, J. Convex programming for disjunctive convex optimization. **Mathematical Programming**, [S.l.], v.86, n.3, p.595–614, dec 1999.

[13] CORNUÉJOLS, G. Valid inequalities for mixed integer linear programs. **Mathematical Programming**, [S.l.], v.112, n.1, p.3–44, 2008.

[14] DOWSON, O.; KAPELEVICH, L. SDDP.jl: a Julia package for Stochastic Dual Dynamic Programming. **Optimization Online**, [S.l.], 2017.

[15] GOMORY, R. E. An algorithm for integer solutions to linear programs. **Recent advances in mathematical programming**, [S.l.], v.64, p.260–302, 1963.

[16] GUROBI OPTIMIZATION, I. **Gurobi Optimizer Reference Manual**. 2016.

[17] JEROSLOW, R. Cutting-Plane Theory: disjunctive methods. In: **Studies in Integer Programming**. [S.l.]: Elsevier, 1977. p.293–330.

[18] JEROSLOW, R. G. Representability in mixed integer programmiing, I: characterization results. **Discrete Applied Mathematics**, [S.l.], v.17, n.3, p.223–243, jun 1987.

[19] KAPELEVICH, L. **SDDiP.jl**: SDDP extension for integer local or state variables. 2018.

[20] PEREIRA, M. V.; PINTO, L. M. Multi-stage stochastic optimization applied to energy planning. **Mathematical programming**, [S.l.], v.52, n.1-3, p.359–375, 1991.

[21] PHILPOTT, A. B.; GUAN, Z. On the convergence of stochastic dual dynamic programming and related methods. **Operations Research Letters**, [S.l.], v.36, n.4, p.450–455, 2008.

[22] ROCKAFELLAR, R. T. **Convex Analysis (Princeton Landmarks in Mathematics and Physics)**. [S.l.]: Princeton University Press, 1996.

[23] RUSZCZYNSKI, A. **Stochastic Programming Handbook In Operations Research and Management Science - Vol. 10**. [S.l.]: Elsevier, 2003.

[24] SHAPIRO, A. Analysis of stochastic dual dynamic programming method. **European Journal of Operational Research**, [S.l.], v.209, n.1, p.63–72, 2011.

[25] SHAPIRO, A. **Tutorial on risk neutral, distributionally robust and risk averse multistage stochastic programming**. Optimization Online.

[26] SHAPIRO, A.; DENTCHEVA, D.; RUSZCZYŃSKI, A. **Lectures on stochastic programming**: modeling and theory. [S.l.]: SIAM, 2009.

[27] THOMÉ, F. S. **Representação de não-convexidades no planejamento da operação hidrotérmica utilizando PDDE**. 2013. Tese (Doutorado em Ciência da Computação) — Federal University of Rio de Janeiro/COPPE.

[28] ZOU, J.; AHMED, S.; SUN, A. **Stochastic Dual Dynamic Integer Programming**. preprint Optimization Online.