HABILITATION À DIRIGER DES RECHERCHES

Discipline : Mathématiques

Présentée par

Vincent Leclère

CERMICS, École des Ponts ParisTech

Exact methods and applications of optimization under uncertainty

---

Soutenue le 17 Avril 2023 devant le jury composé de

Mme Marianne Akian	INRIA et École Polytechnique	Rapporteure
M. Daniel Kuhn	École Polytechnique Fédérale de Lausanne	Rapporteur
M. Andrezj Ruszczyński	Rutgers University	Rapporteur
M. Jean-Philippe Chancelier	École des Ponts	Examinateur
M. Welington de Oliveira	École des Mines de Paris	Examinateur
M. Stéphane Gaubert	INRIA et École Polytechnique	Examinateur

À Apolline.

Que je suis impatient de t'emmener chasser,
crayon à la main, dans ces contrées merveilleuses
qui naissent de quelques axiomes.

Remerciements

La rédaction de ce manuscrit m'a donné l'occasion de faire le point sur la première partie de ma carrière académique et je voudrais remercier ici, sans prétendre à l'exhaustivité, ceux et celles qui ont participé à cette aventure.

En premier lieu, je tiens à très chaleureusement remercier mes rapporteurs, Marianne Akian, Daniel Kuhn et Andrezj Ruszczyński. J'ai un immense respect pour chacun d'entre vous et je suis très honoré que vous ayez pris le temps de lire mes travaux. Un grand merci aussi pour les autres membres du jury, Jean-Philippe qui n'avait pu être dans mon jury de thèse, Welington avec qui j'espère pouvoir développer une collaboration plus active, et Stéphane pour ses conseils et son expérience.

Je voudrais remercier aussi tous les collègues, du laboratoire et d'ailleurs, avec qui j'ai pu travailler, échanger, enseigner ou co-encadrer ces dernières années. Parmi vous tous je voudrais remercier tout particulièrement ceux qui m'ont apporté leur expérience: Pierre pour sa patience, Michel et Jean-Philippe pour leur présence continue, Frédéric pour sa disponibilité et ses conseils avisés, Eric et Tony pour leur soutien et leur écoute ces dernières années, Stéphane pour son puits sans fond de savoirs et d'expériences, Andy pour avoir toujours renouvelé les opportunités de collaboration malgré la distance. Merci aussi à ceux qui grandissent avec moi dans le monde académique, et en particulier à Axel pour avoir partagé bureau et projets ; à Bernardo pour s'être retrouvé sur les chemins de la convexité ; à Christelle pour être une amie avant tout ; à Jérémy pour me faire découvrir de nouveaux défis ; à Laurent pour m'avoir le premier expliqué SDDP.

Je souhaite ensuite remercier les doctorants que j'ai pu accompagner ces dernières années et dont on retrouvera une partie des travaux ici. Henri, merci pour ta bonne humeur permanente ; François, merci pour ton sérieux en particulier en ce qui concerne les aspects numériques ; Étienne, merci pour avoir toujours su rapprocher les questions théoriques des aspects pratiques ; Maël, merci pour ta remarquable capacité à transformer de vagues idées en théorèmes. Je n'oublie pas Regan et Carlos, venus faire un post-doc ici. Vous encadrer fut toujours un échange enrichissant, tant sur le plan mathématique qu'humain. Rassurez-vous, vos successeurs (Zoé, Louis, Vitor...) m'apportent tout autant de questions, de réponses, de surprises ou de paragraphes à retravailler que vous.

Il me faut bien évidemment remercier l'École et plus particulièrement le CERMICS pour la qualité de vie et de travail qui y règne. Merci à tous les collègues pour le soutien quotidien, l'écoute, l'entraide et la bonne humeur générale. Un merci tout particulier à Isabelle et Stéphanie qui nous simplifient la vie à tellement d'égards qu'on se demande si le CERMICS restera debout le jour de leur départ.

Finalement, merci à mes proches : amis de longue dates (Adrien, Balthazar, Cyril, Manu, Guillaume, Lionel, Marina, Martin, Raphaël...) qui acceptent parfois de m'écouter divaguer autour de la convexité cachée ou de congestion dans les réseaux de transports ; parents, qui m'ont donné le goût des maths en vendant le chien ou me donnant des matrices à manipuler, et soutenu tout au long de mon parcours. Et enfin, merci à Anne-Christine, qui partage ma vie depuis 10 ans, m'écoute patiemment lui présenter de nouveaux objets récemment découverts ; râler contre élèves ou rapporteurs ; ou faire la cobaye pour un sujet d'examen. Merci à Apolline, d'être venue raccourcir nos nuits, enrichir nos vies, et me pousser à enfin terminer ce manuscrit, que je suis impatient de t'expliquer.

Contents

Organization of the manuscript

This manuscript sums up some of my work in optimization under uncertainty as a researcher at École des Ponts from 2014 to 2022.

The manuscript is made of a preliminary chapter introducing concepts, notation and challenges of optimization under uncertainty, and two parts of roughly equal importance, the first being dedicated to theoretical results, and the second to various applications of optimization under uncertainty. We end the manuscript with some perspectives.

We present the main results and some proof arguments, but the reader is referred to the original papers for technical details and more computational results.

Part I. Exact methods in multistage stochastic optimization

This part mainly considers multistage problems with uncertain parameters modeled as stagewise independent random variables with known law, paving the way to Dynamic Programming methodologies. If those random variables have non-discrete distributions, the traditional approach consists in sampling them first, and then optimizing over the sampled problem, using statistical theory to characterize the results obtained with some confidence.

Chapter 2 is based on the Ph.D. thesis work of Maël Forcier co-supervised with Stéphane Gaubert, described in [VL13, VL12]. It leverages geometric tools to obtain, in the linear case, a discretization method that yields the same value as the original problem. This discretization can either be universal (that is working for all probability law) or not, and uniform (that is valid for all first-stage decisions) or local (that is for a given first-stage decision). In the second case, it is used as part of adapted partition methods to provide an effective algorithm for two-stage linear problems with non-finitely supported noises.

Chapters 3 and 4 are the result of a long-term work on the Stochastic Dual Dynamic Programming (SDDP) algorithm, with various colleagues. In Chapter 3, we present a generic framework, called Trajectory Following Dynamic Programming, encompassing the well-known SDDP and multiple variants, as well as convergence results based on [VL3, VL12]. These algorithms work by progressively refining (exact or error-controlled) lower approximations of cost-to-go functions. If lower bounds, or more precisely outer bounds, are common, computing inner bounds for multistage stochastic problems, even under the finitely supported noise assumption, is challenging. Consequently, Chapter 4 leverages duality theory to compute outer bounds of the dual, and thus inner bounds of the original problem. It results first from a collaboration with Jean-Philippe Chancelier, Pierre Carpentier and François Pacaud [VL10], and then with Bernardo da Costa [VL14].

Part II. Applications of optimization under uncertainty

This part presents various applications of decision under uncertainty problems.

Chapter 5 mainly stems from a collaboration with Pierre Carpentier, Jean-Philippe Chancelier, Michel De Lara and François Pacaud [VL2, VL7]. It presents decomposition methods for multistage stochastic problems, with a special focus on spatial decomposition tools mainly motivated by energy-related problems. It also leverages epi-convergence results from [VL9].

Chapter 6 results from a project with Andy Philpott during Henri Gerard's Ph.D. thesis, and discusses risked competitive partial equilibrium for risk-averse agents with strictly concave objective functions presented in [VL8]. In contrast with social planning problems, or equilibrium, with risk-neutral agents we show that risked equilibrium are not unique.

Finally, Chapter 7 covers some work done during the Ph.D. thesis of Étienne de Saint-Germain supervised by Frédéric Meunier and myself. It discusses applications of optimization under uncertainty in supply chain on lot-sizing problems, especially on the trade-off between costs, stocks and service level. This work is presented in [VL6].

Chapters	Papers
Chapter 2	[VL13, VL12]
Chapter 3	[VL3, VL16]
Chapter 4	[VL10, VL14]
Chapter 5	[VL2, VL7, VL9]
Chapter 6	[VL8]
Chapter 7	[VL6]
Not discussed	[VL1, VL4, VL5, VL11, VL15]

Table 1: Overview of papers discussed in the manuscript

List of publications

[VL1] Matheus Grasselli, Vincent Leclère, and Mike Ludkovski. Priority option: the value of being a leader. *International Journal of Theoretical and Applied Finance*, 16(01):1350004, 2013.

[VL2] Michel De Lara, Pierre Carpentier, Jean-Philippe Chancelier, and Vincent Leclère. Optimization methods for the smart grid. *Report commissioned by the Conseil Français de l'Energie, Ecole des Ponts ParisTech*, 2014.

[VL3] Pierre Girardeau, Vincent Leclère, and Andrew B Philpott. On the convergence of decomposition methods for multistage stochastic convex programs. *Mathematics of Operations Research*, 40(1):130–145, 2015.

[VL4] Michel De Lara and Vincent Leclère. Building up time-consistency for risk measures and dynamic optimization. *European Journal of Operational Research*, 249(1):177–187, 2016.

[VL5] Eugene Ndiaye, Olivier Fercoq, Alexandre Gramfort, Vincent Leclère, and Joseph Salmon. Efficient smoothed concomitant lasso estimation for high dimensional regression. In *Journal of Physics: Conference Series*, volume 904, page 012006. IOP Publishing, 2017.

[VL6] Etienne de Saint Germain, Vincent Leclère, and Frédéric Meunier. A stochastic multi-item lot-sizing problem with bounded number of setups. In Greg H. Parlier, Federico Liberatore, and Marc Demange, editors, *Proceedings of the 7th International Conference on Operations Research and Enterprise Systems, ICORES 2018, Funchal, Madeira - Portugal, January 24-26, 2018*, pages 106–114. SciTePress, 2018.

[VL7] Pierre Carpentier, Jean-Philippe Chancelier, Vincent Leclère, and François Pacaud. Stochastic decomposition applied to large-scale hydro valleys management. *European Journal of Operational Research*, 270(3):1086–1098, 2018.

[VL8] Henri Gérard, Vincent Leclère, and Andy Philpott. On risk averse competitive equilibrium. *Operations Research Letters*, 46(1):19–26, 2018.

[VL9] Vincent Leclère. Epiconvergence of relaxed stochastic optimization problems. *Operations Research Letters*, 47(6):553–559, 2019.

[VL10] Vincent Leclère, Pierre Carpentier, Jean-Philippe Chancelier, Arnaud Lenoir, and François Pacaud. Exact converging bounds for stochastic dual dynamic programming via fenchel duality. *SIAM Journal on Optimization*, 30(2):1223–1250, 2020.

[VL11] Axel Parmentier, Victor Cohen, Vincent Leclère, Guillaume Obozinski, and Joseph Salmon. Integer programming on the junction tree polytope for influence diagrams. *INFORMS Journal on Optimization*, 2(3):209–228, 2020.

[VL12] Maël Forcier and Vincent Leclère. Generalized adaptive partition-based method for two-stage stochastic linear programs: Geometric oracle and analysis. *Operations Research Letters*, 50(5):452–457, 2022.

[VL13] Maël Forcier, Stéphane Gaubert, and Vincent Leclère. Exact quantization of multistage stochastic linear problems. *arXiv preprint arXiv:2107.09566*, 2021.

[VL14] Bernardo Freitas Paulo da Costa and Vincent Leclère. Dual sddp for risk-averse multistage stochastic programs. *arXiv preprint arXiv:2107.10930*, 2021.

[VL15] Jeremy Bleyer and Vincent Leclère. Robust limit analysis theory for computing worst-case limit loads under uncertainties. *arXiv preprint arXiv:2203.11354*, 2022.

[VL16] Maël Forcier and Vincent Leclère. Convergence of trajectory following dynamic programming algorithms for multistage stochastic problems without finite support assumptions. *Optimization-online*, (8946), 2022.

Notation

We are going to use the following notational convention:

- Random variables are written in bold, their realization in normal font
- $[\![a, b]\!]$ represents the set of integer between a and b, and $[n] = [\![1, n]\!]$
- $(\Omega, \mathcal{A}, \mathbb{P})$ represents a probability space, where Ω is the sample space, $\mathcal{A}$ a σ-algebra, and $\mathbb{P}$ a probability measure
- $\overline{\mathbb{R}} \coloneqq \mathbb{R} \cup \{-\infty, +\infty\}$
- If A is a finite set, $|A|$ is its cardinality
- $f^\star$ denotes the Fenchel conjugate of f (see Eq. (4.7))
- $\boldsymbol{x}_t \preccurlyeq \mathcal{A}_t$ stands for the random variable $\boldsymbol{x}_t$ being measurable with respect to the σ-algebra $\mathcal{A}_t$
- $\mathbb{E}_{\mathbb{P}}$ stands for the expectation with respect to (the often omitted) probability $\mathbb{P}$
- $\mathcal{B}$ (resp. $\mathcal{F}$) are the backward (resp. forward) Bellman operators (see Section 1.3)
- MSP is a shorthand for multistage stochastic problem
- $\mathrm{conv}(A)$ is the convex hull of set A, and $\mathrm{ri}(A)$ its relative interior
- $\mathrm{epi}(f)$ is the epigraph of function f, $\mathrm{dom}(f)$ its domain
- $\mathrm{supp}(\boldsymbol{\xi})$ is the support of random variable $\boldsymbol{\xi}$

We end with some standard notation: x denotes the state of a system, u a control, ξ an exogenous uncertain variable, ρ a risk-measure. The notation $\hat{V}_t$ usually refers to a cost-to-go, or Bellman, function, and V_t its expected (or risk-adjusted) counterpart. Most of the time, $\overline{V}_t$ (resp. $\underline{V}_t$) are upper bound (resp. lower bound) of V_t. Finally, $\mathcal{B}$ usually refers to a (backward) Bellman operator.

Chapter 1

An introduction to optimization under uncertainty

This chapter is a didactic, but incomplete, presentation of optimization under uncertainty: its definitions, concepts, challenges and methodologies. It serves as an introduction to the rest of the manuscript as well as an opportunity to fix notation and definitions.

1.1 How to approach a decision under uncertainty problem?

Imagine that you have an industrial optimization problem and want to take uncertainty on some parameters into account. In this Section, we go through a series of questions one should ask when trying to add uncertainty to some deterministic problem: What is your risk preference? What are the constraints? What is the information structure? We end with an important practical recommendation: set up a simulator to evaluate any solution proposed.

Remark 1.1 (Where does uncertainty come from?)**.** *Uncertainty can be due to lack of knowledge (*e.g., *missing data or private information), lack of computational power (*e.g., *high resolution weather forecast model cannot be run over large area and long horizon), truly random parameters*[2] *(*e.g., *number of clients on a given day), forecasting or computing errors, actions of others, etc.*

1.1.1 Representing risk preference

Modeling decision-making under uncertainty is a challenge in itself as it is difficult to represent *uncertain* parameters ξ. Indeed, with an uncertain parameter, you now have to find a decision $x \in X$ minimizing an objective $J(x, \xi)$ parametrized by an unknown ξ. The question is, how to determine which of $\xi \mapsto J(x_1, \xi)$ or $\xi \mapsto J(x_2, \xi)$ is preferable? In fact, as there is no natural total ordering on functional spaces, this is a subjective and modeling choice, as illustrated by Fig. 1.1.

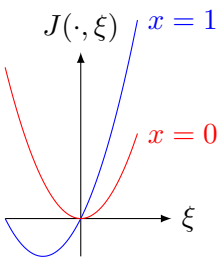


Figure 1.1: Parametrized cost function $J(x, \cdot) : \xi \mapsto (\xi + x)^2 - x$ for two values of x.

[2] Although one might philosophically argue that most of what we casually consider random, like a die throw, is actually deterministic but woefully chaotic, and cannot be forecasted due to lack of data and/or computing power.

There are two main paradigms: robust and stochastic optimization. Robust optimization essentially assumes that the uncertain parameter ξ belongs in a known *uncertainty set* R, and is chosen in an adversarial way. On the other hand, stochastic optimization assumes that $\boldsymbol{\xi}$ is a random variable (hence the bold font), with known law. Bridging both approaches is the *distributionally robust optimization* paradigm that assumes that $\boldsymbol{\xi}$ is a random variable whose probability law is chosen adversarially from a set $\mathcal{Q}$.

In abstract form, the problems we study are formulated as

$$\operatorname*{Min}_{x\in X} \rho(J(x,\boldsymbol{\xi})), \tag{1.1}$$

where x is the decision variable, J the objective function (including constraints), and ξ represents the uncertain variable. Here, ρ is an operator that maps functions of ξ, parametrized by x, into $\overline{\mathbb{R}}$.

On the one hand, *Robust optimization*, with uncertainty set R, considers the following choice of ρ:

$$(RO) \qquad \rho(J(x,\boldsymbol{\xi})) = \sup_{\xi\in R} J(x,\xi). \tag{1.2}$$

On the other hand, a classical operator ρ for *stochastic optimization* would be, for some probability measure $\mathbb{P}$,

$$(SO) \qquad \rho(J(x,\boldsymbol{\xi})) = \mathbb{E}_{\mathbb{P}}\Big[J(x,\boldsymbol{\xi})\Big] \coloneqq \int_{\Xi} J(x,\xi)\,\mathrm{d}\mathbb{P}(\xi). \tag{1.3}$$

When ρ is actually an expectation with respect to some probability, as in (1.3), we say that we have a *risk-neutral* stochastic optimization problem. Aggregators such as ρ are known as *risk measures*; see *e.g.,* [ADEH99, RS06b, PP12].

Definition 1.2 (risk measure)**.** *Let $(\Omega,\mathcal{A},\mathbb{P})$ be a probability space, and $L^0(\Omega,\mathcal{A},\mathbb{P};\mathbb{R}\cup\{+\infty\})$ the set of measurable random variables with values in $\mathbb{R}\cup\{+\infty\}$. We call any function that maps $L^0(\Omega,\mathcal{A},\mathbb{P};\mathbb{R}\cup\{+\infty\})$ into $\overline{\mathbb{R}}$ a* risk measure.

Further, we say that[1] *ρ is*

1. monotone *if $\boldsymbol{X}\leqslant\boldsymbol{Y}\Longrightarrow\rho\big[\boldsymbol{X}\big]\leqslant\rho\big[\boldsymbol{Y}\big]$;*
2. translation-equivariant *if $\rho(\boldsymbol{X}+c)=\rho\big[\boldsymbol{X}\big]+c$ for all $c\in\mathbb{R}$;*
3. convex *if it is monotone, translation-equivariant and satisfies*

$$\rho\big[t\boldsymbol{X}+(1-t)\boldsymbol{Y}\big]\leqslant t\rho\big[\boldsymbol{X}\big]+(1-t)\rho\big[\boldsymbol{Y}\big],\qquad\forall t\in[0,1],\quad\forall\boldsymbol{X},\boldsymbol{Y}.$$

4. positive homogeneous *if, for all $t>0$, $\rho\big[t\boldsymbol{X}\big]=t\rho\big[\boldsymbol{X}\big]$;*
5. law invariant *if, for any two identically distributed random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$, we have $\rho\big[\boldsymbol{X}\big]=\rho\big[\boldsymbol{Y}\big]$.*

Finally, a coherent risk measure *is a monotone, translation-equivariant, convex, positive homogeneous and law-invariant risk measure.*

A Fenchel-Moreau convex analysis result allows, with some light topological requirement,[2] to give a dual representation of the coherent risk measures.

[1] Unfortunately the community has not been able to agree on a single sign convention for these definitions. I choose here to adopt the convention that $\boldsymbol{X}$ represents an uncertain cost—to be minimized, and $\rho\big[\boldsymbol{X}\big]$ represents a certain equivalent, meaning a determinist cost considered equivalent to the uncertain one.

[2] locally convex paired topological space with a technical condition on the coupling, satisfied by all L^q spaces coupled with their topological dual.

Theorem 1.3 (Thm 2.2 [RS06b])**.** *Let $p \in [1, +\infty]$, and $\rho : L^p(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{R} \cup \{+\infty\}) \to \overline{\mathbb{R}}$ be a proper, lower semicontinuous,*[1] *convex function. Then ρ is a coherent risk measure if and only if there exists a set $\mathcal{Q}$ of probability measure such that, for all $\boldsymbol{X} \in L^p(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{R} \cup \{+\infty\})$, we have*

$$\rho\big[\boldsymbol{X}\big] = \sup_{\mathbb{Q} \in \mathcal{Q}} \mathbb{E}_{\mathbb{Q}}\big[\boldsymbol{X}\big].$$

Further $\mathcal{Q}$ can be chosen as a closed convex set.[2]

The most well-known, and largely used law-invariant coherent risk measures is the Average Value at Risk.[3]

Definition 1.4 (Average Value at Risk (AVAR))**.** *Let $\boldsymbol{X}$ be an integrable, real-valued random variable. Then its* value at risk *of level $\alpha \in (0, 1)$ is defined as*

$$VaR_\alpha = \min\{z \in \mathbb{R} \mid F_X(z) \geqslant \alpha\}, \tag{1.4}$$

where $F_X(z) \coloneqq \mathbb{P}(\boldsymbol{X} \leqslant z)$. Then its average value at risk of level α is given by one of the following equivalent formulas.

$$AVaR_\alpha = \frac{1}{1-\alpha} \int_{[\alpha,1)} VaR_\beta(\boldsymbol{X}) \, \mathrm{d}\beta \tag{1.5a}$$

$$= \min_{s \in \mathbb{R}} \left(s + \frac{1}{1-\alpha} \mathbb{E}\Big[(X - s)^+\Big] \right) \tag{1.5b}$$

$$= \max_{\mathbb{Q} \in \mathcal{Q}} \mathbb{E}_{\mathbb{Q}}[X]. \tag{1.5c}$$

with $\mathcal{Q} = \Big\{$probability measures $\mathbb{Q}$ such that $\left\| \frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}_X} \right\|_\infty \leqslant \frac{1}{1-\alpha} \Big\}$

Further, if $\mathbb{P}(X = VaR_\alpha) = 0$, then

$$AVaR_\alpha = \mathbb{E}[X \mid X \geqslant VaR_\alpha]. \tag{1.5d}$$

The 4 formulas for the AVAR have different interpretations and use cases. Consider that $\boldsymbol{X}$ is a random cost. Equation (1.5a) might be seen as the original definition of AVAR. Equation (1.5b) is introduced by Rockafellar and Uryasev in [RU$^+$00] as an efficient way to estimate the AVAR (and VaR, given by the optimal s). It is a useful reformulation from an optimization perspective, allowing to cast risk-averse problems as risk-neutral with a new optimization variable. Equation (1.5c) explicitly gives the dual formulation of AVAR. Equation (1.5d) might be the most intuitive interpretation of AVAR: $AVAR_\alpha(\boldsymbol{X})$ is the expected value of the cost $\boldsymbol{X}$ knowing that we are in the $1 - \alpha$ worst cases.

There are other widely used coherent risk measures. For example, a great choice for both modeling and mathematical properties is a convex combination of expectation and AVAR

$$\rho(\boldsymbol{X}) = (1 - \theta)\mathbb{E}[\boldsymbol{X}] + \theta AVAR_\alpha(\boldsymbol{X}).$$

Another good choice is upper semideviation [OR99, OR01], for $\alpha \in [0, 1]$,

$$\rho(\boldsymbol{X}) \coloneqq \mathbb{E}[\boldsymbol{X}] + \alpha \mathbb{E}\Big[(\max(0, \boldsymbol{X} - \mathbb{E}[\boldsymbol{X}])^p\Big]^{1/p}.$$

Remark 1.5 (Distributionally Robust Optimization)**.** *There is a third, recent and very active paradigm that bridges robust and risk-neutral stochastic optimization:* Distributionally Robust Optimization (DRO)*. The aggregator ρ in DRO assumes that $\boldsymbol{\xi}$ is a random variable whose law $\mathbb{P}$ is chosen adversarially in*

[1] Equivalent to the Fatou condition required by some authors.

[2] more precisely choose $\mathcal{Q} = \mathrm{dom}(\rho^\star)$.

[3] Commonly called Conditional Value at Risk (CVAR) which might be an ill-fated choice when we want to consider conditional risk measures. Is also known as Tail Value at Risk (TVAR), Expected Shortfall (ES) or superquantile.

a set $\mathcal{Q}$. Thus, if $\mathcal{Q}$ contains all Diracs in R, it is equivalent to a robust optimization problem; if it contains a single probability measure, it is equivalent to a risk-neutral stochastic optimization approach. Finally, mathematically speaking[1]*, DRO is equivalent to risk-averse stochastic optimization with (lower semicontinuous) coherent risk measure, as both consider a supremum of expectations against multiple probability measures.*

1.1.2 Uncertainty and constraints

To illustrate the difficulty of modeling constraint under uncertainty, consider the following toy problem:

$$\begin{aligned} \underset{(u_t,h_t)_{t\in[T]}}{\text{Min}} \quad & \sum_{t=1}^{T} h_t \\ \text{s.t.} \quad & h_t \geqslant \underline{h}_t && \forall t \in [T], \\ & h_t = h_{t-1} + u_t && \forall t \in [T], \\ & |u_t| \leqslant \Delta && \forall t \in [T]. \end{aligned}$$

where $h_0 = 0$, which can model the altitude of a drone, with fixed ground speed, going from a point A at time 0 to a point B at time T while needing to stay above the ground (of altitude $\underline{h}_t$) to avoid crashing, with some constraint over the vertical speed. Now, if the drone vertical speed control is imprecise and has an error of about σ_0, one might be inclined (with very good theoretical and practical reasons) to model the dynamics of the system as

$$\boldsymbol{h}_t = \boldsymbol{h}_{t-1} + u_t + \boldsymbol{\xi}_t, \qquad \forall t \in [T],$$

where $\boldsymbol{\xi}_t$ is a centered Gaussian of standard deviation σ_0, and the constraint is satisfied almost surely, that is,

$$\mathbb{P}(\boldsymbol{h}_t \geqslant \underline{h}_t) = 1, \qquad \forall t \in [T].$$

Unfortunately, this would lead to an empty set of feasible solutions. Indeed, whatever the current altitude h_t and (bounded) vertical speed chosen there is a positive probability, albeit potentially very small,[2] of crashing.

In this example, we saw that uncertainty in constraints can easily lead to an empty set of feasible solutions, in which case any optimization method will either be useless or provide a solution of questionable meaning.

There are alternatives to almost sure constraints, among which are *chance constraints*, which require a solution to satisfy a constraint with at least some probability; *penalization*, which replaces the hard constraint by a cost of not satisfying it[3]; *recourse variables* which allow correcting one's decision after observing the uncertain parameter.

Remark 1.6 (Chance constraint)**.** *A* chance constraint *is a constraint of the form*

$$\mathbb{P}[g(x, \boldsymbol{\xi}) \geqslant 0] \geqslant 1 - \varepsilon, \tag{CC}$$

where x is the decision variable and $\boldsymbol{\xi}$ the uncertain parameter modeled as a random variable.

This type of constraint has drawn a lot of interest in the optimization community, especially since it is seemingly intuitive: if x satisfies (CC)*, then it has at least probability $1 - \varepsilon$ to be satisfied for the actual (unknown) value of ξ.*

[1] They are different in the way the set $\mathcal{Q}$ is chosen: in risk-averse optimization, it is given a priori as a representation of the subjective risk-sensitivity of the decision maker, while in DRO it is a set representing our lack of knowledge of the true probability law $\mathbb{P}$, so typically given as a ball around an empirical distribution.

[2] A security margin of $10\sigma_0$ yields a probability of crashing lower than $7.5\,10^{-24}$, which is *very* small, but still positive.

[3] Most non-physical constraints are actually soft constraints and can be accurately modeled by an adequate cost.

*Unfortunately, this type of constraint also has its downsides. First, it is mathematically difficult to deal with (*e.g., *one can lose the convexity of the admissible set of solutions without adequate assumptions). Second, it can be misunderstood by practitioners. In particular, chance constraints do not distinguish, for a given scenario $g(x, \xi)$, between violating the constraint by a small or a large margin (*i.e., $g(x, \xi) > 0$ *small or not), when, in most practical cases, there is a practical difference between the two.*

1.1.3 Information structure

The question of known information when making (part of) a decision is crucial in optimization under uncertainty. Intuitively, there is a difference between a decision that we are committed to from the start (sometimes called *here-and-now* decisions), and *recourse* (sometimes called *wait-and-see*) decisions that can adapt to the uncertainty.

In multistage stochastic (or robust) optimization the question is even more complex as the uncertainty is progressively revealed, meaning that some decisions are taken with no information[1] on the uncertainty, some with part of the information, and some with full information. Those *information constraints* are often modeled through measurability constraints (or scenario trees) as formally introduced in Section 1.2.

Remark 1.7 (Uncertainty law modeling)**.** *One of the most unsettling points for newcomers to stochastic optimization might be the need for an uncertainty law. To understand this point let's denote by $(\boldsymbol{\xi}_t)_{t\in[T]}$ our uncertainty variables,* e.g., *representing demand at time t for a given product.*

Practitioners are often required to produce the "best prediction", which in our case would be a forecast of demand $(\overline{\xi}_t)_{t\in[T]}$. But a single prediction leads to deterministic optimization, and overfitting of the solution to this prediction. Thus, in addition to the "best prediction" (often the expected value) $(\overline{\xi}_t)_{t\in[T]}$ we need some model of "prediction error" $\boldsymbol{\varepsilon}_t \coloneqq \boldsymbol{\xi}_t - \overline{\xi}_t$.

This can be thought, at least in the two-stage setting, as requiring quantiles of $(\boldsymbol{\xi}_t)_{t\in[T]}$. However, in multistage optimization, quantiles are not enough: we also need the stagewise dependence of the error $(\varepsilon_t)_{t\in[T]}$. In other words, one has to answer the following question: If, at time t, the actual demand ξ_t is above the predicted demand $\overline{\xi}_t$, does the actual demand at time $t+1$ have more chance of being above $\overline{\xi}_{t+1}$ (because the product is more appreciated than forecast), or under it (because more people have already satisfied their need) or neither?

Even harder, one's best prediction $(\overline{\xi}_t)_{t\in[T]}$, is probably updated as time passes. Properly modeling a multistage stochastic optimization requires one to model this update, that is determine how one's prediction is evolving knowing one's past errors. Finding data and fitting a model for that is always challenging.

1.1.4 Of the importance of a simulator

Multistage optimization under uncertainty problems are difficult from a modeling, theoretic and practical point of view. When faced with a practical problem in this class, there are two routes. One can look for heuristics (*e.g.,* reinforcement learning) with few to no convergence guarantee. Or one can simplify the problem, solve the simplified version through dedicated methods and then reconstruct a solution to the original problem. In the end, this is also a (mat)heuristic.

We could compare this need with (large-scale) combinatorial optimization, which is also often numerically intractable and require some (mat)heuristics. There is, however, a key difference: in combinatorial optimization, the admissibility of a solution and its associated costs are often easy to compute. For example, consider multidimensional bin-packing: checking admissibility and evaluating the cost of a candidate solution is straightforward. This is not the case in decision under uncertainty: evaluating cost is often subjective (especially regarding risks, see § 1.1.1), and can be numerically challenging, as it often requires at least to compute an expectation. Checking admissibility is even worse.

[1] Except its distribution law / uncertainty set.

Remark 1.8 (Monte Carlo simulation)**.** *Even in the discrete case, computing an expectation - which includes probability of events - can be a numerically challenging task. Indeed, if we have* 10 *possible realization per week over a week, then computing the expectation consists in doing a sum over* 10^{52} *values, which is roughly a hundred times the number of atoms on earth.*

Fortunately, probability theory provides a simple way to make precise statistical estimations. Indeed, if $\boldsymbol{C}$ *is a real-valued random variable with finite variance* σ^2*,* e.g.*, the cost induced by a given solution of a multistage program, we can always estimate its expectation by its empirical average using the law of large number, the error made being controlled through the central limit theorem.*

More precisely, if we draw N_{MC} *independent realizations of* $\boldsymbol{C}$*, denoted* $(C_n)_{n\in[N_{MC}]}$*, then denoting the average cost* $\overline{C}_{N_{MC}} \coloneqq \dfrac{1}{N_{MC}} \sum_{n\in[N_{MC}]} C_n$*, we have that,*

$$\mathbb{P}\Big(\mathbb{E}[\boldsymbol{C}] \in \Big[\overline{C}_{N_{MC}} - \beta\frac{\sigma}{\sqrt{N_{MC}}}\,,\, \overline{C}_{N_{MC}} + \beta\frac{\sigma}{\sqrt{N_{MC}}}\Big]\Big) \xrightarrow[N_{MC}\to\infty]{} \mathbb{P}(\boldsymbol{G} \in [-\beta,\beta]),$$

where $\boldsymbol{G}$ *is a centered reduced Gaussian law. If the standard deviation* σ *is unknown but can be replaced by a convergent estimator (for example the empirical standard deviation). In practice, we often use* $\beta = 1.96$ *which yields a confidence of* 95%*.*

That is why it is important when considering a decision under uncertainty problem, to decide on a *simulator* and a scenario generation procedure. Scenario generation can be done by using historical data or a (maybe complex) stochastic model and produces possible realizations of the uncertain parameters.[1] The simulator in itself is a bit of computer code that takes as input multiple uncertainty realizations, a *policy* (*i.e.,* a function that uses available information to make a decision), and applies it to compute the associated costs and various other Key Performance Indicators. The simulator is, in the end, what decides which heuristic or simplification yields the best result for the problem at hand.

1.2 Multistage stochastic optimization

Multistage stochastic optimization is concerned with problems where the set of admissible decisions X is a set of stochastic processes. More precisely, we want to model a problem where the decision maker makes a first decision x_1, then observes some part of the uncertainty $\boldsymbol{\xi}_1$ before making a second decision x_2, and observing a second part of the uncertainty $\boldsymbol{\xi}_1$ and so on.

We start by giving some technical background, before detailing the two main approaches that are used to tackle such problems.

1.2.1 Technical set-up

In Problem (1.1) we did not detail the set X, and the reader might have implicitly assumed that it was a subset of $\mathbb{R}^n$ so that x represents a deterministic decision. To address multistage stochastic problem we consider that X is a set of stochastic processes, or equivalently a set of measurable functions.

Consider a probability space $(\Omega, \mathcal{A}, \mathbb{P})$. Let $\{\boldsymbol{\xi}_t\}_{t\in[T]}$ be a sequence of random variables with support $\Xi_t \coloneqq \operatorname{supp}(\boldsymbol{\xi}_t)$. For notational consistency, we also define a deterministic initial condition as a random variable $\boldsymbol{x}_0$ whose support is $\operatorname{supp}(\boldsymbol{x}_0) = \{x_0\}$ and further assume that $\operatorname{supp}(\boldsymbol{\xi}_1) = \{\xi_1\}$. We consider the following[2] (risk-neutral) *Multistage Stochastic Program* (MSP)

[1] Including predictions of primitive parameters, as well as their updates.

[2] We assume here that distribution and function are chosen such that the expectation is defined (with potentially non-finite value).

$$\underset{(\boldsymbol{x}_t)_{t\in[T]}\in L^0(\Omega,\mathcal{A},\mathbb{P};\mathbb{R}^{nT})}{\text{Min}} \quad \mathbb{E}_{\mathbb{P}}\left[\sum_{t=1}^{T} c_t(\boldsymbol{x}_{t-1},\boldsymbol{x}_t,\boldsymbol{\xi}_t) + V_{T+1}(\boldsymbol{x}_T)\right] \tag{1.6a}$$

$$\text{s.t.} \quad (\boldsymbol{x}_{t-1},\boldsymbol{x}_t) \in P_t(\boldsymbol{\xi}_t) \qquad \forall t\in[T], \tag{1.6b}$$

$$\boldsymbol{x}_t \preccurlyeq \mathcal{A}_t \coloneqq \sigma(\boldsymbol{\xi}_1,\dots,\boldsymbol{\xi}_t) \qquad \forall t\in[T]. \tag{1.6c}$$

where constraint (1.6b) is the dynamic constraint that links together decision at time t and $t+1$, *i.e.,* $P_t(\xi_t)$ is the set of possible transition for the state at stage t under realization ξ_t, and constraint (1.6c) is an information constraint, the so-called *non-anticipativity* constraint, that states that the decision variable $\boldsymbol{x}_t$ is measurable with respect to the past noises $(\boldsymbol{\xi}_{[t]})$. By the Doob–Dynkin lemma, Problem (1.6), where the decision variables are random processes, can be equivalently written in the following functional form

$$\underset{(\phi_t)_{t\in[T]}}{\text{Min}} \quad \mathbb{E}_{\mathbb{P}}\left[\sum_{t=1}^{T} c_t(\boldsymbol{x}_{t-1},\boldsymbol{x}_t,\boldsymbol{\xi}_t) + V_{T+1}(\boldsymbol{x}_T)\right] \tag{1.7a}$$

$$\text{s.t.} \quad (\boldsymbol{x}_{t-1},\boldsymbol{x}_t) \in P_t(\boldsymbol{\xi}_t) \qquad \forall t\in[T], \tag{1.7b}$$

$$\boldsymbol{x}_t = \phi_t(\boldsymbol{\xi}_1,\dots,\boldsymbol{\xi}_t) \qquad \forall t\in[T]. \tag{1.7c}$$

where the decision variable is now a vector of measurable functions ϕ_t mapping the past noises realizations into $\mathbb{R}^n$.

Remark 1.9 (Explicit Control)**.** *Problem* (1.6) *is presented in a state-only formulation with an implicit dynamic implied by constraint* (1.6b)*.*

In some settings it is more natural and practical to consider a controlled stochastic dynamical system*, that is assuming that the sequence of random variables $(\boldsymbol{x}_t)_{t\in[T]}$ satisfies*

$$\boldsymbol{x_t} = \mathrm{dyn}_t(\boldsymbol{x}_{t-1},\boldsymbol{u}_t,\boldsymbol{\xi}_t) \qquad \forall t\in[T], \tag{1.8}$$

where dyn *is the* dynamics *of the system, $\boldsymbol{x}$ is the* state*, $\boldsymbol{u}$ is the control and $\boldsymbol{\xi}$ the noise.*

With explicit controls, a MSP is formulated as:

$$\underset{(\boldsymbol{u}_t)_{t\in[T]}\in L^0(\Omega,\mathcal{A},\mathbb{P};\mathbb{R}^{nT})}{\text{Min}} \quad \mathbb{E}_{\mathbb{P}}\left[\sum_{t=1}^{T} L_t(\boldsymbol{x}_{t-1},\boldsymbol{u}_t,\boldsymbol{\xi}_t) + V_{T+1}(\boldsymbol{x}_T)\right] \tag{1.9a}$$

$$\textit{s.t.} \quad \boldsymbol{x_t} = \mathrm{dyn}_t(\boldsymbol{x}_{t-1},\boldsymbol{u}_t,\boldsymbol{\xi}_t) \qquad \forall t\in[T] \tag{1.9b}$$

$$\boldsymbol{u}_t \in \mathcal{U}_t(\boldsymbol{x}_{t-1},\boldsymbol{\xi}_t) \tag{1.9c}$$

$$\boldsymbol{u}_t \preccurlyeq \mathcal{A}_t \qquad \forall t\in[T]. \tag{1.9d}$$

We formally recover Problem (1.6) *by setting*

$$c_t(x_{t-1},x_t,\xi_t) = \inf\left\{L_t(x_{t-1},u_t,\xi_t) \quad | \quad u_t\in\mathcal{U}_t(x_{t-1},\xi_t), \quad x_t = \mathrm{dyn}_t(x_{t-1},u_t,\xi_t)\right\}, \tag{1.10a}$$

$$P_t(\xi) = \left\{(x_{t-1},x_t) \quad | \quad \exists u_t\in\mathcal{U}_t(x_{t-1},\xi_t), \quad x_t = \mathrm{dyn}_t(x_{t-1},u_t,\xi_t)\right\}. \tag{1.10b}$$

Going from Problem (1.6) *to Problem* (1.9) *is also possible with measurable selection theorems. Thus, overlooking the technical details, we can say that both formulations, with or without explicit control, have the same modeling power. For a given application one might feel more natural than the other. In this manuscript, we generally choose the implicit control formulation mainly for notational conciseness reasons.*

Remark 1.10 (Final cost V_{T+1})**.** *In Problem* (1.6)*, we assumed that we are in a finite horizon setting. However, some problems do not naturally have a finite horizon T, or a final cost function V_{T+1}. To*

address these problems we usually resort to using methods for infinite horizon problems, often with discounted costs (see, e.g., *[Ber12]).*

*Alternatively, we need to define a relevant V_{T+1}, which represents the cost associated with the final state of the system. There are various elements one can consider when modeling V_{T+1}: it can be seen as the cost of dismantling the project once it has reached its end; it can be used as a way of inducing the final state to be in a reasonable place (*e.g., *a quadratic cost to push the final state toward a nominal value); it can represent the cost of managing the system after V_{T+1}.*

In the last case, T does not represent the actual end of the system, but only the horizon considered for practical reasons. Let's illustrate a simple procedure to derive a reasonable final cost V_{T+1} in this case. For the sake of clarity consider the problem of managing a system over one year ($T = 365$), we also assume that x_0 is an admissible state for $T = 365$. We can solve the one-year problem with $V^0_{T+1} = 0$, for all initial states reachable at time T. We can solve again the one-year problem, but using this time as final cost $V^1_{T+1} = V^0_0 - V_0(x_0)$. The solution obtained is also an optimal solution over a two-year problem.[1] *We can repeat the process until two successive end-of-horizon costs are close enough,* i.e., $\|V^{k+1}_{T+1} - V^k_{T+1}\| \leqslant \varepsilon$.

Having set up the optimization problems we now present solution approaches.

1.2.2 Stochastic programming

In its simplest form, the stochastic programming (SP) approach makes the following assumption.

Assumption (FSN) (Finitely supported noise)**.** *The support of the random process $(\boldsymbol{\xi}_t)_{t\in[T]}$ is finite.*

The noise can then be described through a *scenario tree*.

Definition 1.11 (scenario tree)**.** *Let $(\boldsymbol{\xi}_t)_{t\in[T]}$ be a sequence of finitely supported random variables, with $\mathrm{supp}(\boldsymbol{\xi}_t) = \Xi_t$. The associated* scenario tree[2] *$\mathcal{T}$ is defined as follows.*

$n_0 = \{\emptyset\}$ is the root node of $\mathcal{T}$. We construct $\mathcal{N}_t$, the set of nodes of $\mathcal{T}$ of depth t, by induction, starting with $\mathcal{N}_0 = \{n_0\}$. For $t \in [T-1]$, and $n \in \mathcal{N}_t$, the set of children of n is given as $\big\{(n,\xi) \mid \xi \in \Xi_{t+1}\big\}$. Further, we define $\pi_n = \mathbb{P}((\boldsymbol{\xi}_\tau)_{\tau\in[t]} = n)$.

The leaf nodes of $\mathcal{T}$ are called the scenarios *of the tree and denoted by $\mathcal{L}$.*

Thus,[3] a node $n \in \mathcal{N}_t$ is defined as the collection $\xi_{[t]} = (\xi_1, \dots, \xi_t)$ of the noises realizations up to time t.

With this definition the functional formulation of MSP (1.7) simply consists in associating a decision variable to each node of the tree, leading to what is known as the *extended formulation*:

$$\underset{(x_n)_{n\in\mathcal{T}}}{\mathrm{Min}} \quad \sum_{t=1}^{T-1}\Big\{\sum_{n\in\mathcal{N}_{t-1}}\sum_{\xi\in\Xi_t}\pi_{(n,\xi)}L_t(x_n, x_{(n,\xi)}, \xi)\Big\} + \sum_{m\in\mathcal{L}}\pi_m V_{T+1}(x_m) \tag{1.11a}$$

$$\text{s.t.} \quad (x_n, x_{(n,\xi)}) \in P_t(\xi) \qquad \forall \xi \in \Xi_{t+1}, \quad \forall n \in \mathcal{N}_{t-1}, \quad \forall t \in [T]. \tag{1.11b}$$

Thus, this problem is a large structured deterministic problem, that can either be solved through standard deterministic software or dedicated methods like the L-Shaped methods [VSW69, Bir85] or Progressive Hedging [RW91].

The main drawback of this approach is that the number of variables in the extensive formulation is proportional to the number of nodes in $\mathcal{T}$ which is exponential in the number of stages T, and highly sensitive to the *branching size i.e.,* the size of the support of the noises Ξ_t.

[1] The optimal value differ by $V_0(x_0)$.

[2] Another, equivalent, construction of the scenario tree is given in § 5.1.1.

[3] Up to straightforward abuse of notation.

Remark 1.12 (Sample Average Approximation)**.** *Without the finite noise Assumption (FSN), or if the number of scenarios is too large, we usually proceed by sampling. Note that Chapter 2 presents, in the linear setting, a way of obtaining exact, non-statistical, discretizations.*

More precisely we simulate, independently, N realizations $(\xi^s)_{s\in[N]}$ of $\boldsymbol{\xi}$, and replace the true probability $\mathbb{P}$ by the empirical law, i.e., *the uniform (discrete) law on $(\xi^s)_{s\in[N]}$. In other words, we solve the sampled problem instead of the original one.*

For a two-stage problem, a uniform law of large number ensures that, under light assumptions, this sample average approximation *converges toward the original problem in various sense. We refer to [SDR14, Chap. 5] for details.*

For multistage problem sampling is more complicated, as sampling $\boldsymbol{\xi}_{[T]}$ in bulk does not represent correctly the information structure of the problem. For example, with continuous distributions, there is no reason to have two different sampled scenario having the same second-stage noise, meaning that every decision for $t > 1$ has full knowledge of the noise realization on the sampled problem. The solution consists in using conditional sampling: first, draw N realizations of $\boldsymbol{\xi}_2$, then draw N realizations of $\boldsymbol{\xi}_3$ conditionally on each realization of $\boldsymbol{\xi}_2$, etc. With this conditional sampling, we again obtain convergence results, but at the price of an exponential number of scenarios.

1.2.3 Dynamic Programming

To tackle problems with a large (or even infinite) number of steps we need some assumption on the stochastic process in order to compress the information required to make an optimal decision. Indeed, we have seen that, in the generic setting of Problem (1.6), an optimal decision $\boldsymbol{x}_t^\sharp$ at time t is a function of all past noises $(\boldsymbol{\xi}_{[t]})$. Dynamic Programming relies on finding a sufficient statistic $\boldsymbol{s}_t$, called the *state*, for $\boldsymbol{x}_t^\sharp$, *i.e.,* on describing the optimal solution $\boldsymbol{x}_t^\sharp$ as a function of $\boldsymbol{s}_t$.

This can be done by using the structure of the problem's constraints (see Eq. (1.6b)), and making some sort of Markovian assumption on $(\boldsymbol{\xi}_t)_{t\in[T]}$. We thus introduce the following assumption, that holds for most of this manuscript.

Assumption (SWI) (Stagewise independence)**.** *$(\boldsymbol{\xi}_t)_{t\in[T]}$ is a sequence of independent exogeneous random variables,* i.e., *such that the law of $\boldsymbol{\xi}_t$ is independent of all decisions variables.*

Under Assumption (SWI), $\boldsymbol{s}_t = (\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)$ is a state of Problem (1.7) which is equivalent to

$$\min_{(\psi_t)_{t\in[T]}} \quad \mathbb{E}_{\mathbb{P}}\left[\sum_{t=1}^{T} c_t(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t, \boldsymbol{\xi}_t) + V_{T+1}(\boldsymbol{x}_T)\right], \tag{1.12a}$$

$$\text{s.t.} \quad (\boldsymbol{x}_{t-1}, \boldsymbol{x}_t) \in P_t(\boldsymbol{\xi}_t) \qquad \forall t \in [T], \tag{1.12b}$$

$$\boldsymbol{x}_t = \psi_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t) \qquad \forall t \in [T]. \tag{1.12c}$$

Remark 1.13 (Limited memory)**.** *The stagewise independence of noises is one of the main limits of the Dynamic Programming approach. This can be relaxed by extending the state. Indeed, if $(\boldsymbol{\xi}_t)_{t\in[T]}$ is (part of) a Markov Chain, then it is enough to consider the state $\boldsymbol{s}_t = (\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_{t-1}, \boldsymbol{\xi}_t)$. More generally if $(\boldsymbol{\xi}_t)_{t\in[T]}$ has limited memory of depth ℓ,*[1] *then we can consider the state $\boldsymbol{s}_t = (\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_{t-\ell}, \dots, \boldsymbol{\xi}_t)$.*

We now introduce the set of admissible next states,

$$\mathcal{X}_t(x_{t-1}, \xi) \coloneqq \big\{x_t \mid (x_{t-1}, x_t) \in P_t(\xi)\big\} \qquad \forall \xi \in \Xi_t, \quad \forall t \in [T], \tag{1.13}$$

and the collection of reachable sets

$$X_0^r = \{x_0\}, \tag{1.14a}$$

$$X_t^r = \bigcup_{x_{t-1}\in X_{t-1}^r} \bigcup_{\xi\in\Xi_t} \mathcal{X}_t(x_{t-1}, \xi) \qquad \forall t \in [T]. \tag{1.14b}$$

[1] That is $\boldsymbol{\xi}_t$ is independent of $(\boldsymbol{\xi}_\tau)_{\tau\in[t-\ell-1]}$ conditionally to $(\boldsymbol{\xi}_\tau)_{\tau\in[\![t-\ell,t-1]\!]}$.

Leveraging Assumption (SWI), we can rewrite Problem (1.12) in a nested form

$$\begin{aligned}\operatorname*{Min}_{x_1\in\mathcal{X}_1(x_0,\xi_1)} c_0(x_0,x_1,\xi_1)+\mathbb{E}_{\mathbb{P}}\Bigg[\operatorname*{Min}_{\boldsymbol{x}_2\in\mathcal{X}_1(x_1,\boldsymbol{\xi}_2)} c_1(x_1,\boldsymbol{x}_2,\boldsymbol{\xi}_2)+\mathbb{E}_{\mathbb{P}}\Bigg[\dots \\ +\mathbb{E}_{\mathbb{P}}\Bigg[\operatorname*{Min}_{\boldsymbol{x}_T\in\mathcal{X}_T(\boldsymbol{x}_{T-1},\boldsymbol{\xi}_T)} c_T(\boldsymbol{x}_{T-1},\boldsymbol{x}_T,\boldsymbol{\xi}_T)+V_{T+1}(\boldsymbol{x}_T)\Bigg]\Bigg],\Bigg]\end{aligned} \tag{1.15}$$

which, coupled with Assumption (SWI), leads to a recursive equation, known as Bellman's equation:

$$V_t(x_{t-1})=\mathbb{E}_{\boldsymbol{\xi}_t}\Big[\inf_{\boldsymbol{x}_t\in\mathcal{X}_t(x_{t-1},\boldsymbol{\xi}_t)} L_t(x_{t-1},\boldsymbol{x}_t,\boldsymbol{\xi}_t)+V_{t+1}(\boldsymbol{x}_t)\Big],\qquad V_{T+1}\text{ given.} \tag{1.16}$$

In the fully finite case, *i.e.,* when, for all $t\in[T]$, the reachable set X_t^r, the admissible next-state $\mathcal{X}_t(\cdot,\cdot)$, and noise support Ξ_t are finite set, we can easily use the above Bellman equation to solve the MSP problem. This is described in Algorithm 1. In particular, we can see that solving MSP through DP requires $O(T\times\overline{X^r}\times\overline{\mathcal{X}}\times\overline{\Xi})$ elementary operations, where the overline stands for an upper bound on the cardinality of associated sets, for example $|\Xi_t|\leqslant\overline{\Xi}$ for all $t\in[T]$.

Data: Transition costs c_t, final cost V_{T+1},
Result: Optimal policy $(\psi_t)_{t\in[T]}$ and cost-to-go functions $(V_t)_{t\in[T]}$;
for $t:T\to 0$ **do**
  $V_t\equiv 0$
  for $x\in X_t^r$ **do**
    for $\xi\in\Xi_t$ **do**
      $\hat{V}_t(x,\xi)=\infty$;
      for $\widetilde{x}\in\mathcal{X}_t(x,\xi)$ **do**
        $v_{\widetilde{x}}=c_t(x,\widetilde{x},\xi)+V_{t+1}(\widetilde{x})$;
        if $v_{\widetilde{x}}<\hat{V}_t(x,\xi)$ **then**
          $\hat{V}_t(x,\xi)=v_{\widetilde{x}}$;
          $\psi_t(x,\xi)=\widetilde{x}$;
      $V_t(x)\leftarrow V_t(x)\mathbb{P}(\boldsymbol{\xi}_t=\xi)+\hat{V}_t(x,\xi)$

Algorithm 1: Stochastic Dynamic Programming algorithm - finite case

Remark 1.14 (Dynamic programming with explicit control)**.** *Consider the set-up of Problem* (1.9)*, with stagewise independent noise (Assumption (SWI)). Assume that, at time $t\in[T]$, X_t is the set of possible states, and U_t the set of possible controls. Then Algorithm 1 can be straightforwardly adapted as follows.*

Data: *Loss functions L_t, final cost V_{T+1},*
Result: *Optimal policy $(\psi_t)_{t\in[T]}$ and cost-to-go functions $(V_t)_{t\in[T]}$;*
for $t:T\to 0$ **do**
  $V_t\equiv 0$
  for $x\in X_t$ **do**
    for $\xi\in\Xi_t$ **do**
      $\hat{V}_t(x,\xi)=\infty$;
      for $u\in\mathcal{U}_t(x,\xi)\subseteq U_t$ **do**
        $\widetilde{x}=\mathrm{dyn}(x,u,\xi)$;
        $v_u=L_t(x,u,\xi)+V_{t+1}(\widetilde{x})$;
        if $v_u<\hat{V}_t(x,\xi)$ **then**
          $\hat{V}_t(x,\xi)=v_u$;
          $\psi_t(x,\xi)=u$;
      $V_t(x)\leftarrow V_t(x)+\mathbb{P}(\boldsymbol{\xi}_t=\xi)\hat{V}_t(x,\xi)$

Algorithm 2: Stochastic Dynamic Programming algorithm - dynamical system formulation

Remark 1.15 (Curse(s) of dimensionality)**.** *Algorithm 1 and Algorithm 2 underline the curses of dimensionality faced by the Dynamic Programming approach to MSP: its time complexity is linear in the number of states at time t which is often exponential in the dimension of the state vector.*

More precisely, assume that the MSP problem has d_x independent dynamical systems, each with n_x possible values at time t, d_u independent controls, each with n_u possible values at time t, and d_ξ independent noises each with n_ξ possible values. Then the number of elementary operations required by Algorithm 1 is $O(T \times n_x^{d_x} \times n_u^{d_u} \times n_\xi^{d_\xi})$. This illustrates the three curses of dimensionality*:*

state *which is the most commonly thought about, and most difficult to contend with;*

control *or equivalently admissible next state, which is due to brute force minimization, and can be limited by using more advanced minimization tools;*

noise *which can always be reduced by estimating the expectations using Monte Carlo methods.*

Remark 1.16 (Stochastic Programming or Dynamic Programming?)**.** *The Stochastic Programming (SP) and Dynamic Programming (DP) approaches both tackle MSPs, but their numerical limits are quite different.*

On one hand, SP is numerically limited by the horizon T and branching size $|\Xi_t|$ and relies on our capacity to solve large-scale deterministic problems (in particular linear problems), but the state vector dimension is not very impactful, and no stagewise independence assumption on the noises is required.

On the other hand, DP complexity is linear in the horizon $[T]$ and branching size, but exponential in state dimension and requires some Markovian assumption for the noises. However, the algorithm is made of elementary operations and does not rely on any deterministic solver. [1]

Further, SP mainly provides an estimation of the value of the problem, as well as a good first-stage control. DP mainly provides cost-to-go function estimation, hence the estimation of the value of the problem as well as policies.

In a nutshell: stochastic programming is adapted to problems with small horizon T of 3 or 4 and specific form (typically linear), while dynamic programming is adapted to problems with independent noise and small state dimension (3 or 4). These limits are illustrated in Example 1.17.

Example 1.17 (Illustrating the numerical limits)**.** *To illustrate the numerical limits of both approaches, we consider the problem of managing a hydroelectric valley made of 7 interconnected dams, over a year, with a weekly time-step. There is one state and one control per dam, and the random variable is the inflows, assumed to be stagewise independent and discretized in 10 values per time-step.*

An SP approach requires a scenario tree of depth 52 leading to 10^{52} scenarios. It is currently estimated that the global internet will have 2×10^{23} bytes of stored data by 2025. Thus, we will never be able to write, let alone pass to a solver, the extended formulation of Problem (1.11) with 52 time steps.

On the other hand, a DP approach where each state is discretized into 100 points and each control is discretized into 10 points leads to $O(52 \times 10 \times 100^7 \times 10^7) \approx 5 \times 10^{22}$ floating point operations. As of November 2021, the fastest supercomputer in the TOP500 (Fugaku) boast 4.42×10^{17} floating point operations per second. We would need around 2 days of computation on it to solve this problem by Algorithm 2 with Fugaku, and 3 million years for a 10 dams problem.

This problem is easily solved to reasonable precision through algorithms like Stochastic Dual Dynamic Programming that are discussed in Part I of the manuscript.

1.2.4 Some settings of interest

We present here some extensions and specifications of the MSP problem (1.6) that are of interest in this manuscript.

[1] Although the TFDP algorithms of Chapter 3, tends to address problems with larger state dimension by using more structure and relying on solvers.

First, to extend the scope of Problem (1.6), consider a multistage risk measure (see [RS06a]) ϱ which associate to a sequence of random cost $(\boldsymbol{C}_t)_{t\in[T+1]}$ a real number. To simplify things, we are only considering recursive multistage risk measures of the form

$$\varrho((\boldsymbol{C}_t)_{t\in[T+1]}) = \rho_{\boldsymbol{\xi}_1}\Big(\boldsymbol{C}_1 + \rho_{\boldsymbol{\xi}_2|\boldsymbol{\xi}_{[1]}}\Big(\boldsymbol{C}_2 + \cdots + \rho_{\boldsymbol{\xi}_T|\boldsymbol{\xi}_{[T-1]}}\big(\boldsymbol{C}_T + \rho_{\boldsymbol{\xi}_{T+1}|\boldsymbol{\xi}_{[T]}}(\boldsymbol{C}_{T+1})\big)\Big)\Big). \quad (1.17)$$

where each $\rho_{\boldsymbol{\xi}_{t+1}|\boldsymbol{\xi}_t}$ is a (coherent) conditional risk measure in the sense of [RS06a].

Then, the risk-averse MSP problem reads

$$\begin{aligned}
\underset{(\boldsymbol{x}_t)_{t\in[T]}}{\text{Min}} \quad & \varrho\Big(c_1(\boldsymbol{x}_0,\boldsymbol{x}_1,\boldsymbol{\xi}_1), c_2(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{\xi}_2),\dots,c_T(\boldsymbol{x}_{T-1},\boldsymbol{x}_T,\boldsymbol{\xi}_T), V_{T+1}(\boldsymbol{x}_T)\Big) && \text{(1.18a)}\\
\text{s.t.} \quad & (\boldsymbol{x}_{t-1},\boldsymbol{x}_t) \in P_t(\boldsymbol{\xi}_t) && \forall t\in[T], \ \text{(1.18b)}\\
& \boldsymbol{x}_t \preccurlyeq \mathcal{A}_t \coloneqq \sigma(\boldsymbol{\xi}_1,\dots,\boldsymbol{\xi}_t) && \forall t\in[T]. \ \text{(1.18c)}
\end{aligned}$$

Definition 1.18 (risk-neutral / convex / linear settings)**.** *Problem* (1.18) *is said to be* nested *if the multistage risk measure is of the form* (1.17)*.*

Problem (1.18) *is said to be* risk-neutral *if, for all* $t\in[T]$*,* $\rho_{\boldsymbol{\xi}_{t+1}|\boldsymbol{\xi}_{[t]}}(\boldsymbol{X})$ *is the conditional expectation* $\mathbb{E}\big[\boldsymbol{X}\big|\,\boldsymbol{\xi}_{[t]}\big]$*.*

Problem (1.18) *is said to be* convex *if it is nested and, for all* $t\in[T]$*, and all relevant* ξ*, i)* $c_t(\cdot,\cdot,\xi)$ *is (jointly) convex,* $P_t(\xi)$ *is a closed convex set, and iii)* V_{T+1} *is convex.*

Problem (1.18) *is said to be* linear*, denoted MSLP, if it is risk-neutral and for all* $t\in[T]$*, and all relevant* ξ*, i)* $c_t(\cdot,\cdot,\xi)$ *is linear,* $P_t(\xi)$ *is a polyhedron, and iii)* V_{T+1} *is polyhedral.*

It is also interesting to discuss some recourse settings that are sometimes useful for algorithms. Recall that X_t^r, defined in (1.14) represents the set of states that can be attained at time t by satisfying the constraints up to time t, but with no guarantee after that. To guarantee that all reachable states can be part of an admissible trajectory, we introduce the notion of *relatively complete recourse*.

Definition 1.19 (Relatively Complete Recourse)**.** *We say that Problem* (4.5) *satisfy a relatively complete recourse (RCR) assumption if, for all* $t\in[T]$*, all* $x_{t-1}\in X_{t-1}^r$*, all* $\xi\in\operatorname{supp}(\boldsymbol{\xi}_t)$*, there exists an admissible next state* $x_t\in\mathcal{X}_t(x_{t-1},\xi)$ *such that* $c_t(x_{t-1},x_t,\xi)<\infty$*.*

Remark 1.20 (Non-decomposable contraints)**.** *According to our current convention, the above constraint* (1.18b) *is implicitely understood as almost sure, with* $P_t(\boldsymbol{\xi}_t(\omega))$ *being a deterministic set. In other words, in a splitted extensive formulation*[1] *of problem* (1.18)*, constraint* (1.18b) *is duplicated independently for each scenario.*

However, in some cases, constraint (1.18b) *should be allowed to couple scenarios. One use-case would be to model (conditional) expectation constraints, asking that the conditional value of* $\boldsymbol{x}_t$ *knowing* $\boldsymbol{x}_t$ *belongs to some deterministic set. This appears naturally when using duality, as in Chapter 4. Another use-case would be to model measurability constraints, to allow for finer information representation than* (1.18c)*.*

Finally, for the sake of completeness, recall that Assumption (SWI) ensures that the noises are stagewise independent, while Assumption (FSN) ensures that the noises are finitely supported.

1.3 Bellman operators

In order to discuss more advanced DP approaches, and most results of Part I, we now introduce some Bellman operators on cost-to-go functions. The Backward Bellman operator is a way of approximating cost-to-go functions over horizon $[\![t,T]\!]$ from an approximate cost-to-go function over horizon $[\![t+1,T]\!]$. The operator takes as argument a function $\widetilde{V}_{t+1}$ and returns a function $\widetilde{V}_t$. Forward Bellman operators form a method of determining the best next state given the current state, noise realization and approximated cost-to-go. Where backward Bellman operators return approximated cost-to-go functions, forward Bellman operators return admissible policies.

[1] see Chapter 5 for definitions

1.3.1 Backward Bellman operator

For a measurable proper l.s.c function $\widetilde{V} : \mathbb{R}^{n_t} \to \mathbb{R} \cup \{+\infty\}$, we denote the Bellman operator of Problem (1.12) applied to $\widetilde{V}$ by

$$\hat{\mathcal{B}}_t(\widetilde{V}) = \begin{cases} \mathbb{R}^{n_{t-1}} \times \Xi_t & \to \overline{\mathbb{R}} \\ (x_{t-1}, \xi_t) & \mapsto \inf\limits_{x_t \in \mathcal{X}(x_{t-1}, \xi_t)} c_t(x_{t-1}, x_t, \xi_t) + \widetilde{V}(x_t) \end{cases}, \tag{1.19a}$$

and

$$\mathcal{B}_t(\widetilde{V}) : x_{t-1} \mapsto \mathbb{E}\big[\hat{\mathcal{B}}_t(\widetilde{V})(x_{t-1}, \boldsymbol{\xi}_t)\big]. \tag{1.19b}$$

With this notation, the Bellman equation (1.16) reads

$$V_t = \mathcal{B}_t(V_{t+1}) \qquad \forall t \in [T], \qquad V_{T+1} \text{ given}, \tag{1.20}$$

and the value of Problem (1.12) is simply $V_1(x_0)$.

Remark 1.21 (Convex/Linear Bellman operator)**.** *A large part of the results presented here rely on a convex or linear setting introduced in § 1.2.4.*

We say that $\mathcal{B}_t$ is a Convex Bellman operator *(CBO) if, for almost all ξ, $c_t(\cdot, \cdot, \xi)$ is a (jointly) convex function, and $P_t(\xi)$ is convex. In particular, for every convex function $\widetilde{V}$, evaluating $\hat{\mathcal{B}}_t(\widetilde{V})$ is a convex optimization problem.*

A CBO $\mathcal{B}_t$ is a Linear Bellman operator *(LBO) if, for almost all ξ, $c_t(\cdot, \cdot, \xi)$ is a (jointly) convex function, and the graph of $P_t(\xi)$ is polyhedral. In particular, for every polyhedral function $\widetilde{V}$, evaluating $\hat{\mathcal{B}}_t(\widetilde{V})$ is a linear optimization problem.*

Remark 1.22 (risk-averse backward Bellman operators)**.** *A classical extension of the Bellman operator defined in Eq.* (1.19b)*, is its risk-averse counterpart. Indeed, consider nested risk-averse problem* (1.18)

$$\begin{aligned} \min_{x_1 \in \mathcal{X}_1(x_0, \xi_1)} c_0(x_0, x_1, \xi_1) + \rho_{2|\boldsymbol{\xi}_1}\Big[\min_{\boldsymbol{x}_2 \in \mathcal{X}_1(x_1, \boldsymbol{\xi}_2)} c_1(x_1, \boldsymbol{x}_2, \boldsymbol{\xi}_2) + \rho_{3|\boldsymbol{\xi}_{[2]}}\Big[\dots \\ + \rho_{T|\boldsymbol{\xi}_{[T-1]}}\Big[\min_{\boldsymbol{x}_T \in \mathcal{X}_T(\boldsymbol{x}_{T-1}, \boldsymbol{\xi}_T)} c_T(\boldsymbol{x}_{T-1}, \boldsymbol{x}_T, \boldsymbol{\xi}_T) + V_{T+1}(\boldsymbol{x}_T)\Big]\Big]\Big]. \end{aligned} \tag{1.21}$$

Then, the backward Bellman operator associated is, as in the risk-neutral case replacing the expectation by ρ_t, that is

$$\mathcal{B}_t(\widetilde{V}) : x_{t-1} \mapsto \rho_t\big[\hat{\mathcal{B}}_t(\widetilde{V})(x_{t-1}, \boldsymbol{\xi}_t)\big]. \tag{1.22}$$

Remark 1.23 (Abstract Bellman operators)**.** *Following Remark 1.20, the backward Bellman operators $\mathcal{B}_t$ that we use are not always given as an aggregation (expectation or risk-adjusted) of parametrized Bellman operators $\hat{\mathcal{B}}_t$. In this case $\mathcal{B}_t(\widetilde{V})$ shall be seen as an aggregated optimization problem (with one decision variable per possible realization of $\boldsymbol{\xi}_t$) instead of the average of disaggregated optimization problem parametrized by ξ_t.*

Without providing details we give two possible reasons.

- *For some problem we can consider a specific information structure (sometimes called decision-hazard-decision structure) where some coordinates of the decision $\boldsymbol{x}_t$ are measurable with respect to $\mathcal{A}_{t-1}$ instead of $\mathcal{A}_t$. In this case, those coordinates are common to all possible realizations of $\boldsymbol{\xi}_t$, leading to an aggregated two-stage problem.*
- *When using duality theory on $\mathcal{B}_t(\widetilde{V})$, as in Chapter 4, we naturally obtain some constraints in expectation, thus preventing disaggregation.*

We can easily derive some useful properties of the (backward) Bellman operators.

Proposition 1.24 (Properties of Bellman operator)**.** *Consider a Bellman operator $\mathcal{B}$, and two canditate cost-to-go functions $\widetilde{V}$ and $\widetilde{R}$. We have the following properties:*

monotonicity *If $\widetilde{V} \leqslant \widetilde{R}$, then $\mathcal{B}(\widetilde{V}) \leqslant \mathcal{B}(\widetilde{R})$.*

translation equivariance *For any constant $k \in \mathbb{R}$, $\mathcal{B}(\widetilde{V} + k) = \mathcal{B}(\widetilde{V}) + k$.*

convexity *Assume that $\mathcal{B}$ is a CBO. If $\widetilde{V}$ is convex, then $\mathcal{B}(\widetilde{V})$ is also convex;*

polyhedrality *Assume that $\mathcal{B}$ is a risk-neutral (abstract) LBO with finitely supported $\boldsymbol{\xi}$. If $\widetilde{V}$ is polyhedral, then $\mathcal{B}(\widetilde{V})$ is also polyhedral.*

1.3.2 Forward Bellman operator and induced policy

Whereas *backward* Bellman operators convert an estimation of the cost-to-go $\widetilde{V}_{t+1}$ from time $t+1$ into an estimation of the cost-to-go from time t, knowing ξ_{t+1}, the *forward* Bellman operator, $\hat{\mathcal{F}}_t$, converts $\widetilde{V}_{t+1}$ into a *policy*. In other words, $\hat{\mathcal{B}}_t(\widetilde{V})$ computes the value of (4.1a), whereas $\hat{\mathcal{F}}_t$ return an optimal solution of (4.1a). In some sense, a choice of forward Bellman operator can be seen as choosing a solver (*e.g.,* for linear or mixed integer linear problems), and a set of parameters, that deterministically returns a solution to a problem of the form (4.1a). We actually need additional technical assumptions, that we do not discuss here, on $\hat{\mathcal{F}}_t$ ensuring, for example, that for all reasonable $\widetilde{V}$ and x, the function $\xi \mapsto \hat{\mathcal{B}}_t(\widetilde{V})(x, \xi)$ is measurable. The interested reader can find a possible set of assumptions in [VL16].

Once we have chosen, for a given problem (1.6), a collection of forward Bellman operators $\{\mathcal{F}_t\}_{t\in[T]}$, any (adequate) approximate collection of cost-to-go functions $(\widetilde{V}_t)_{t\in[T]}$ determines a strategy. Dynamic Programming principle ensures that if $(\widetilde{V}_t)_{t\in[T]}$ is the collection of true cost-to-go functions, then the associated strategy is optimal. Further, any strategy can be *simulated* to estimate an upper bound, see Remark 1.8. This idea of approximating the cost-to-go functions is key in Chapters 3 and 4.

1.4 Going further

In this chapter, we present only some of the possible approaches toward decisions under uncertainty. Most notably we always assumed that the state of the uncertain variable was known. When we only have a *belief* on the state of the uncertainty (or of the state of the system itself), we fall into the class of Partially Observable Markov Decision Problem (POMDP) [Spa12, OA16]. They are a special case of *Influence Diagrams* (see [KF09]) which incorporate more complex information relations.

In a recent stream of papers, W. Powell (see, *e.g.,* [Pow14, Pow19]) tried to classify the various approaches to stochastic optimization. His analysis covers multiple communities and links them to a stochastic optimal control framework (1.12). The crux of the discussion is to explain how different communities provide tractable policies, classifying them in 4 classes: policy function approximations where the policy itself ψ is approximated through a parametrized function (linear, logistics or neural network); cost function approximation where the global cost (present and to come) is approximated; value function approximation where the cost-to-go function is approximated and the policy is given by a forward Bellman operator; and direct look-ahead where one approximate the remainder of the problem and not just the cost-to-go.

Part I

Exact methods in multistage stochastic optimization

Introduction to Part I

In the first part of this manuscript we consider multistage stochastic optimization problems (MSP) of the form (1.6), satisfying the stagewise independence Assumption (SWI), and thus amenable to some Dynamic Programming approaches. Attacking such problems often requires multiple layers of approximations and heuristics. The methods presented in this Part have in common that they are exact approaches that tackle MSPs, by opposition to statistical or heuristics methods.

Often, the first approximation made consists in quantizing the law of the noises $\boldsymbol{\xi}_{[T]}$. Indeed, most stochastic programming or dynamic programming algorithm requires a finite noise assumption Assumption (FSN). The most classical approach is the Sample Average Approximation (SAA) method that samples the costs and constraints. It relies on probabilistic results based on a uniform law of large numbers to provide statistical guarantees, which requires numerous scenarios. Various methodologies aim at reducing the number of scenarios either by quasi-Monte Carlo generation, or by first randomly generating scenarios and then reducing the number with or without guarantees on the approximation made by this reduction. On the other hand, leveraging convexity or monotonicity, we can construct scenario trees that provide lower or upper bounds to the non-finitely supported problem. Chapter 2 takes another path that aims at constructing a finite scenario tree that is equivalent, for the problem considered, to the non-finitely supported distribution. To this aim, we study the higher-order polyhedral structure of linear MSP, using polyhedral geometry tools. This chapter is based on a collaboration with Stéphane Gaubert and our Ph.D. student Maël Forcier, detailed in [VL13, VL12].

Even with a finite, and stagewise independent, noise assumption, solving a MSP is generally intractable and requires approximations. For example, the standard Dynamic Programming algorithm (see Algorithm 1) considers a pre-defined grid of the state-space where the value function are estimated. By contrast, Chapter 3 present a framework of Trajectory Following Dynamic Programming (TFDP) algorithms, (including the well-known and widely used Stochastic Dual Dynamic Programming (SDDP)) that uses an adaptive grid. More precisely, a TFDP algorithm works iteratively, first simulating a state trajectory, and then refining an exact lower approximation of the cost-to-go functions. The framework presented in this chapter was developed for [VL16] building on contributions made in [VL3] and [VL10].

Note that even simply computing an exact upper bound for a MSP is a challenge. Obviously, in the risk-neutral case, once we have chosen an admissible policy, we can always estimate the expected cost by Monte Carlo simulation. For the risk-averse case this approach is rougher, and requires an exponential (in the horizon) number of samples. In any case the upper bound obtained is statistical. Chapter 4 shows how to use duality theory to compute exact, non-statistical, upper bounds in both the risk-neutral and risk-averse case. These results derive from [VL10] and [VL14].

Finally, in [VL11], with A. Parmentier, V. Cohen, G. Obozinski and J. Salmon, we contributed to the field of Influence Diagram where we proposed a MILP formulation. With this approach we can tackle problems with complex information structure (partially observable, weakly coupled, etc.), but assume the set of actions to be discrete and of reasonable size. In addition to an MILP reformulation relying on classical linearization techniques, we proposed valid cuts that derive from the information structure. We do not present this exact method here as the tools used are quite different from the other works.

Chapter 2

Exact quantization of linear stochastic problems

In this chapter, we are considering linear MSPs with not necessarily finitely supported noises. Leveraging polyhedral geometry tools we show how to exactly quantize the MSLP. This chapter is based on a collaboration with Stéphane Gaubert and our Ph.D. student Maël Forcier. This work is detailed in [VL13, VL12].

Roughly speaking, a quantization of an MSLP consists in providing a discrete scenario tree that replaces the original noise distribution. The quantization is said to be *locally exact* at (t, x) if both cost-to-go functions, at time t, of the original and quantized problem are equal at x. The quantization is said to be *uniformly exact* if the value functions of both problems match for all states. Finally, we say that a quantization procedure is *universal* if it does not depend on the distribution of the noises.

2.1 Motivation and setting

Following the definition of § 1.2.4, we consider here a risk-neutral MSLP not necessarily satisfying Assumption (FSN). More precisely, given a sequence of random variables $\boldsymbol{\xi}_t = (\boldsymbol{c}_t, \boldsymbol{\zeta}_t)$, $\boldsymbol{c}_t \in L_1(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{R}^{n_t})$ and $\boldsymbol{\zeta}_t = (\boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)$, indexed by $t \in [T]$, we consider the MSLP given by

$$\begin{array}{lll} \min\limits_{(\boldsymbol{x}_t)_{t\in[T]}} & c_1^\top x_1 + \mathbb{E}\big[\sum\limits_{t=2}^{T} \boldsymbol{c}_t^\top \boldsymbol{x}_t\big] & \\ \text{s.t.} & A_1 x_1 \leqslant b_1, & \\ & \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{x}_{t-1} \leqslant \boldsymbol{b}_t \quad \text{a.s.} & \forall t \in [T], \\ & \boldsymbol{x}_t \in L_\infty(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{R}^{n_t}) & \forall t \in [T], \\ & \boldsymbol{x}_t \preccurlyeq \mathcal{F}_t & \forall t \in [T]. \end{array} \tag{2.1}$$

In this section, we wonder how to deal with general (thus non-finitely supported) distribution in stochastic linear programming. The main question is then

How to solve *exactly* multistage stochastic linear problems with *general distributions* ?

In other words, do there exist analytical formulas and algorithms to compute exact solutions of stochastic programs with general distribution?

Here, we aim at solving exactly the original problem, by finding an equivalent formulation with discrete distributions. This notion of equivalent formulation is best understood through the dynamic programming approach of MSLP. Recall that the *cost-to-go* functions V_t are defined inductively as follows.

We set $V_{T+1} \equiv 0$ and for all $t \in [T]$:

$$\begin{aligned} V_t(x_{t-1}) &:= \mathbb{E}\big[\hat{V}_t(x_{t-1}, \boldsymbol{\xi}_t)\big], \\ \hat{\mathcal{B}}_t(R)(x_{t-1}, \xi_t) &:= \min_{x_t \in \mathbb{R}^{n_t}} \quad c_t^\top x_t + R(x_t) \\ &\qquad \text{s.t.} \quad A_t x_t + B_t x_{t-1} \leqslant b_t, \\ \hat{V}_t &:= \hat{\mathcal{B}}_t(V_{t+1}). \end{aligned} \tag{2.2}$$

where $x_{t-1} \in \mathbb{R}^{n_{t-1}}$, and $\xi_t = (c_t, A_t, B_t, b_t) \in \mathbb{R}^{n_t} \times \mathbb{R}^{q_t \times n_t} \times \mathbb{R}^{q_t \times n_{t-1}} \times \mathbb{R}^{q_t}$.

Note that the results presented here require $(\boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)$ to be finitely supported, while $\boldsymbol{c}_t$ can have a continuous distribution. This separation does not preclude correlation between $\boldsymbol{c}_t$ and $\boldsymbol{\zeta}_t$. However, we require $(\boldsymbol{\xi}_t)\}_{t\in[T]}$ to be a sequence of independent random variables to leverage Dynamic Programming, even though some results can be extended to dependent noises $(\boldsymbol{\xi}_t)_{t\in[T]}$.

Definition 2.1 (Exact quantization)**.** *We say that an MSP (with stagewise independence) admits a* local exact quantization *at time t on x_{t-1} if there exists a finitely supported $(\check{\boldsymbol{\xi}}_t)_{t\in[T]}$ that yields the same expected cost-to-go functions* i.e., *such that*

$$V_t(x_{t-1}) = \mathbb{E}\big[\hat{V}_t(x_{t-1}, \boldsymbol{\xi}_t)\big] = \mathbb{E}\big[\hat{V}_t(x_{t-1}, \check{\boldsymbol{\xi}}_t)\big].$$

In particular, we have $V_t(x_{t-1}) = \sum_{\xi \in \operatorname{supp}(\check{\boldsymbol{\xi}}_t)} \mathbb{P}\big[\boldsymbol{\xi}_t = \xi\big] \hat{V}_t(x_{t-1}, \xi)$.

We call a quantization uniformly exact *if it is locally exact at all $x_{t-1} \in \mathbb{R}^{n_{t-1}}$, and all $t \in [T]$.*

We say that a quantization is partition-based *if there exists a (measurable, finite) partition $\mathcal{P}_{t,x_{t-1}}$ of the uncertainty set Ξ_t such that, for $P \in \mathcal{P}_{t,x_{t-1}}$,*

$$\mathbb{P}\big[\check{\boldsymbol{\xi}}_t = \check{\xi}_{t,P}\big] = \check{p}_{t,P} \quad \textit{with} \quad \check{p}_{t,P} := \mathbb{P}\big[\boldsymbol{\xi}_t \in P\big], \quad \check{\xi}_{t,P} := \mathbb{E}\big[\boldsymbol{\xi}_t \,|\, \boldsymbol{\xi}_t \in P\big]^1.$$

In particular, the partition-based quantization reads

$$V_t(x_{t-1}) = \sum_{P \in \mathcal{P}_{t,x_{t-1}}} \check{p}_{t,P} \hat{V}_t(x_{t-1}, \check{\xi}_{t,P}).$$

	SAA	Jensen/Edmundson-Madansky	Exact quantization
Setting	General	Convex	Linear
Approximations	Confidence interval	Upper and lower bounds	Exact values

Table 2.1: Comparison of advantages and drawbacks of approximation methods with exact quantization

An obvious necessary condition for uniform exact quantization is that the value function V_t is a polyhedral function, meaning that it takes value in $\mathbb{R} \cup \{+\infty\}$ and its epigraph is a (possibly empty) polyhedron. Indeed, for each $\zeta \in \operatorname{supp}(\boldsymbol{\zeta})$, $Q^\zeta : (x, y) \to c^\top y + \mathbb{I}_{Ax+By \leqslant h}$ is polyhedral. Thus, $\hat{V}(\cdot, c, \zeta) := \min_{y \in \mathbb{R}^m} Q^{c,\zeta}(\cdot, y)$ is polyhedral as $\operatorname{epi} \hat{V}(\cdot, \zeta)$ is a projection of $\operatorname{epi} Q^\zeta$. Hence, the following example shows that if the constraints have non-discrete distributions, there is no hope to have a uniform exact quantization theorem. We shall see, however, that this is the case without restrictions on the cost distribution.

Example 2.2 (Stochastic constraints)**.** *In this example, $\boldsymbol{u}$ denotes a uniform random variable on* $[0, 1]$.

[1] When $\check{p}_{t,P} := \mathbb{P}\big[\boldsymbol{\xi}_t \in P\big]$ is equal to 0, $\mathbb{E}\big[\boldsymbol{\xi}_t \,|\, \boldsymbol{\xi}_t \in P\big]$ is not well-defined. Then, we take an arbitrary $\check{\xi}_{t,P}$ in P. This choice does not matter since $\check{\xi}_{t,P}$ will only appear in functions multiplied by $\check{p}_{t,P} = 0$ in formulas.

We first consider a stochastic right-hand side $\boldsymbol{b}$

$$V(x) = \mathbb{E}\begin{bmatrix} \min\limits_{y\in\mathbb{R}^m} & y \\ \textit{s.t.} & \boldsymbol{u} \leqslant y \\ & x \leqslant y \end{bmatrix} = \mathbb{E}\big[\max(x, \boldsymbol{u})\big] = \begin{cases} \frac{1}{2} & \textit{if } x \leqslant 0 \\ \frac{x^2+1}{2} & \textit{if } x \in [0,1] \\ x & \textit{if } x \geqslant 1 \end{cases}. \tag{2.3}$$

We now consider a stochastic constraint matrix $\boldsymbol{B}$*.*

$$V(x) = \mathbb{E}\begin{bmatrix} \min\limits_{y\in\mathbb{R}^m} & y \\ \textit{s.t.} & \boldsymbol{u}x \leqslant y \\ & 1 \leqslant y \end{bmatrix} = \mathbb{E}\big[\max(\boldsymbol{u}x, 1)\big] = \begin{cases} 1 & \textit{if } x \leqslant 1 \\ \frac{x}{2} + \frac{1}{2x} & \textit{if } x \geqslant 1 \end{cases}. \tag{2.4}$$

In both cases the value function is not polyhedral, preventing the existence of a universal *exact quantization.*

We now turn to an example of random recourse matrix $\boldsymbol{W} = 1 + \boldsymbol{u}$*, showing that there does not exist even a* locally *exact partition-based quantization.*

$$\hat{V}(x, u) = \begin{matrix} \min\limits_{y\in\mathbb{R}^m} & y \\ \textit{s.t.} & (1+\boldsymbol{u})y \geqslant 1 \end{matrix} = \frac{1}{u}. \tag{2.5}$$

Thus, $\hat{V}$ *is constant in* x *and convex in* ζ*. In particular, by the law of total expectation and Jensen's inequality for all partition* $\mathcal{P}$ *of* Ξ*:*

$$V(x) = \mathbb{E}\big[\hat{V}(x, \boldsymbol{\zeta})|P\big] = \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big]\mathbb{E}\big[\hat{V}(x, \boldsymbol{\zeta})|P\big] \geqslant \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big]\hat{V}\big(x, \mathbb{E}\big[\boldsymbol{\zeta}|P\big]\big) \tag{2.6}$$

We can then argue by strict convexity that any partition-induced quantization yields a strict upper estimation of the true cost function.

In the remainder of this chapter, we make the main following contributions. In Section 2.2 we briefly introduce some polyhedral geometry tools that are used to derive the results of this section. In Section 2.3 we show that the expected cost-to-go functions are piecewise affine on explicit and universal polyhedral complexes when the constraints are finitely supported and the costs have general distributions. In Section 2.4, we give local and universal exact quantization results for 2SLP with fixed recourse. We also present a generalized adaptive partition-based method (GAPM) to solve 2SLP with fixed recourse but stochastic constraints.

2.2 Polyhedral tools

Our proofs rely on the notions of normal fan and chamber complex of a polyhedron recalled here. These polyhedral objects reveal the geometrical structure of MSLP. Both the normal fan and the chamber complex are special polyhedral complexes. We intend here to make a brief introduction to these objects. A more complete, didactic and self-contained, introduction to polyhedral geometry tools for stochastic optimization can be found in [For22, Chapter 3].

2.2.1 Polyhedral complexes

Polyhedral complexes are collections of polyhedra satisfying some combinatorial and geometrical properties. In particular, the relative interiors of the elements of a polyhedral complex (without the empty set) form a partition of their union. We refer to [DLRS10] for a complete introduction to polyhedral complexes and triangulations.

Definition 2.3 (Polyhedral complex)**.** *A finite collection of polyhedra $\mathcal{C}$ is a* polyhedral complex *if it satisfies i) if $P \in \mathcal{C}$ and F is a non-empty*[1] *face of P then $F \in \mathcal{C}$ and ii) if P and Q are in $\mathcal{C}$, then $P \cap Q$ is a (possibly empty) face of P and Q. Polyhedron in a polyhedral complex are called* cells*. We denote by $\operatorname{supp}\mathcal{C} \coloneqq \bigcup_{P\in\mathcal{C}} P$ the support of a polyhedral complex. Further, if all the elements of $\mathcal{C}$ are polytopes (resp. cones, simplices, simplicial cones), we say that $\mathcal{C}$ is a* polytopal complex *(resp. a* fan*, a* simplicial complex*, a* simplicial fan*).*

We recall that a *simplex* of dimension d is the convex hull of $d+1$ affinely independent point and that a *simplicial cone* of dimension d is the conical hull of d linearly independent vectors.

Proposition 2.4. *For any polyhedral complex $\mathcal{C}$, the relative interiors of its elements (without the empty set) form a partition of its support: $\operatorname{supp}(\mathcal{C}) = \bigsqcup_{P\in\mathcal{C}} \operatorname{ri}(P)$ where the symbol $\sqcup$ denotes a disjoint union.*

For example, the set of faces $\mathcal{F}(P)$ of a polyhedron P is a polyhedral complex.

Definition 2.5 (Refinements and triangulation)**.** *Let $\mathcal{C}$ and $\mathcal{R}$ be two polyhedral complexes, we say that $\mathcal{R}$ is a* refinement *of $\mathcal{C}$, denoted $\mathcal{R} \preccurlyeq \mathcal{C}$, if for every cell $R \in \mathcal{R}$ there exists a cell $C \in \mathcal{C}$ containing R: $R \subset C$.*

Note that $\preccurlyeq$ defines a partial order on the space of polyhedral complexes, and the meet *associated with this order is given by the* common refinement *of two polyhedral complexes $\mathcal{C}$ and $\mathcal{C}'$ defined as the polyhedral complex of the intersections of cells of $\mathcal{C}$ and $\mathcal{C}'$:*

$$\mathcal{C} \wedge \mathcal{C}' \coloneqq \{R \cap R' \mid R \in \mathcal{C}, R' \in \mathcal{C}'\}. \tag{2.7}$$

A triangulation *$\mathcal{T}$ of a polytope Q is a refinement of $\mathcal{F}(Q)$ such that the cells of dimension 0 of $\mathcal{T}$ are the vertices of Q and $\mathcal{T}$ is a simplicial complex. A* triangulation *$\mathcal{T}$ of a cone K is a refinement of $\mathcal{F}(K)$ such that the cells of dimension 1 of $\mathcal{T}$ are the rays of K and $\mathcal{T}$ is a simplicial fan.*

2.2.2 Normal fan

The normal fan is the collection of the normal cones of all faces of a polyhedron. See [LR08] for a review of normal fan properties.

Recall that the *normal cone* of a convex set $C \subset \mathbb{R}^d$ at the point x is the set $N_C(x) \coloneqq \{\alpha \in \mathbb{R}^d \mid \forall y \in C,\ \alpha^\top(y-x) \leqslant 0\}$. More generally, for a set $E \subset C$, $N_C(E) \coloneqq \bigcap_{x\in E} N_C(x)$.

Figure 2.1: Two normally equivalent polytopes P and P' and their normal fan $\mathcal{N}(P) = \mathcal{N}(P')$. The green circle represents the singleton $\{0\}$ which is the normal cone $N_P(x)$ for every $x \in \operatorname{ri}(P)$.

Definition 2.6 (Normal fan)**.** *The* normal fan *of a convex set C is the collection of normal cones*

$$\mathcal{N}(C) \coloneqq \{N_C(x) \mid x \in C\}. \tag{2.8}$$

We say that two convex sets C and C' are normally equivalent *if they have the same normal fan: $\mathcal{N}(C) = \mathcal{N}(C')$, see Fig. 2.1.*

[1] For some authors, a polyhedral complex must contain the empty set. We do not make this requirement.

2.2.3 Chamber complex

The affine regions of the cost-to-go function will correspond to cells of a chamber complex. Projections of polyhedra, fibers and chamber complexes are studied in [BS92, RZ96, Ram96].

Definition 2.7 (Chamber complex)**.** *Let $P \subset \mathbb{R}^d$ be a polyhedron and π a linear projection defined on $\mathbb{R}^d$. For $x \in \pi(P)$ we define the* chamber of x for P along π *as*

$$\sigma_{P,\pi}(x) \coloneqq \bigcap_{F \in \mathcal{F}(P) \text{ s.t. } x \in \pi(F)} \pi(F). \tag{2.9}$$

The chamber complex $\mathcal{C}(P,\pi)$ *of P along π is defined as the (finite) collection of chambers,* i.e.,

$$\mathcal{C}(P,\pi) \coloneqq \{\sigma_{P,\pi}(x) \mid x \in \pi(P)\}. \tag{2.10}$$

Further $\mathcal{C}(P,\pi)$ is a polyhedral complex such that $\operatorname{supp}\mathcal{C}(P,\pi) = \pi(P)$*. In particular,* $\{\operatorname{ri}(\sigma) \mid \sigma \in \mathcal{C}(P,\pi)\}$ *is a partition of $\pi(P)$.*

More generally, the chamber complex of a polyhedral complex $\mathcal{P}$ is

$$\mathcal{C}(\mathcal{P},\pi) \coloneqq \{\sigma_{\mathcal{P},\pi}(x) \mid x \in \pi\big(\operatorname{supp}(\mathcal{P})\big)\}. \tag{2.11}$$

with $\sigma_{\mathcal{P},\pi}(x) \coloneqq \bigcap_{F \in \mathcal{P} \text{ s.t. } x \in \pi(F)} \pi(F)$.

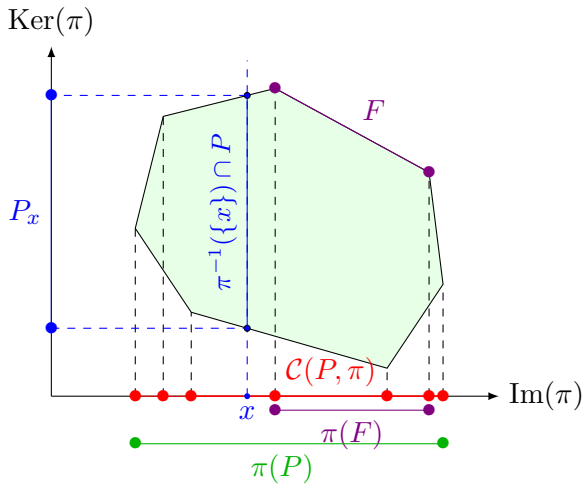


Figure 2.2: A polytope P and its projection in green, its chamber complex in red on the x-axis and a fiber P_x in blue on the y-axis, for the orthogonal projection π on the horizontal axis, a face F and its projection in purple.

Lemma 2.8 (Chamber complex monotonicity with respect to refinement order)**.** *Consider two polyhedral complexes of $\mathbb{R}^d$ and a projection π. If $\mathcal{R} \preccurlyeq \mathcal{S}$ then $\mathcal{C}(\mathcal{R},\pi) \preccurlyeq \mathcal{C}(\mathcal{S},\pi)$.*

Recall that the *fiber* P_x of P along π at x is the projection of $P \cap \pi^{-1}(x)$ on the space $\operatorname{Ker}(\pi)$ (see figure 2.2). An important property of a chamber complex is that all fibers are normally equivalent in each relative interior of cells of the chamber complex. More precisely, let $\sigma \in \mathcal{C}(P,\pi)$ be a chamber, and x and x' two points in its relative interior, then, P_x and $P_{x'}$ are *normally equivalent*, *i.e.,* they have the same normal fan $\mathcal{N}(P_x) = \mathcal{N}(P_{x'})$, see [BS92]. Thus we define the normal fan $\mathcal{N}_\sigma$ above$\sigma \in \mathcal{C}(P,\pi)$ by :

$$\mathcal{N}_\sigma \coloneqq \mathcal{N}(P_x) \quad \text{for an arbitrary } x \in \operatorname{ri}(\sigma). \tag{2.12}$$

2.3 Universal and uniform exact quantization of MSLP with random costs

Leveraging these tools, our aim here is to provide a universal and uniform exact quantization method for MLSP with non-finitely supported random cost. To do so we first study the 2-stage, with deterministic (A_2, B_2, b_2) case before discussing its extension.

2.3.1 Chamber complexes arising from 2-stage problems

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space, $\boldsymbol{c} \in L_1(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{R}^m)$ be an integrable random vector, and assume that (A_2, B_2, b_2) is deterministic. We study the cost-to-go function of the 2-stage stochastic linear problem[1], written as

$$V_2(x) \coloneqq \mathbb{E}\left[\hat{V}_2(x, \boldsymbol{c}_2)\right] \quad \text{with} \quad \hat{V}_2(x, c_2) \coloneqq \min_{y \in \mathbb{R}^m} \; c_2^\top y \quad \text{s.t.} \quad A_2 y + B_2 x \leqslant b_2 \tag{2.13}$$

The dual of the latter problem, for given x and c_2, is

$$\max_{\mu \in \mathbb{R}^q} \; (B_2 x - b_2)^\top \mu \tag{2.14a}$$

$$\text{s.t.} \quad A_2^\top \mu = -c_2, \tag{2.14b}$$

$$\mu \geqslant 0. \tag{2.14c}$$

We denote the *coupling constraint polyhedron* of Problem (2.13) by

$$P \coloneqq \{(x, y) \in \mathbb{R}^{n+m} \mid A_2 y + B_2 x \leqslant b_2\}, \tag{2.15}$$

and π the projection of $\mathbb{R}^n \times \mathbb{R}^m$ onto $\mathbb{R}^n$ such that $\pi(x, y) = x$.

The projection of P is the following polyhedron :

$$\pi(P) = \{x \in \mathbb{R}^n \mid \exists y \in \mathbb{R}^m,\ A_2 y + B_2 x \leqslant b_2\}, \tag{2.16}$$

and for any $x \in \mathbb{R}^n$, the *fiber* of P along π is

$$P_x \coloneqq \{y \in \mathbb{R}^m \mid A_2 y + B_2 x \leqslant b_2\}. \tag{2.17}$$

We first provide a local exact quantization result. Note that, if $x \notin \pi(P)$, then $\hat{V}_2(x, c) = +\infty$, if $x \in \pi(P)$ and $-c \notin \operatorname{cone}(A_2^\top)$, then $\hat{V}_2(x, c) = -\infty$. Assume now that $x \in \pi(P)$ and $-c \in \operatorname{cone}(A_2^\top)$. For each cone $N \in \mathcal{N}(P_x)$, and vector $c_N \in \operatorname{ri}(-N)$, there exists a vector $y_N(x)$ which achieves the minimum in the expression of $\hat{V}_2(x, c_N)$ in (2.13). Further, for any selection of such a $y_N(x)$, we have

$$\hat{V}_2(x, c) = \sum_{N \in \mathcal{N}(P_x)} \mathbb{1}_{c \in -\operatorname{ri} N}\; c^\top y_N(x) \ . \tag{2.18}$$

Assumption 2.1. *The cost $\boldsymbol{c} \in L^1(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{R}^m)$ is integrable with $\boldsymbol{c} \in -\operatorname{cone}(A_2^\top)$ almost-surely.*

Theorem 1 (Uniform quantization of the cost distribution)**.** *Let $x \in \pi(P)$, and σ be a cell of $\mathcal{C}(P, \pi)$ the chamber complex of the coupling constraint polyhedron P along the projection π on the x-space. Assume that $x \in \operatorname{ri}(\sigma)$.*

Under assumption 2.1, for every refinement $\mathcal{R}$ of $-\mathcal{N}_\sigma$, we have:

$$V_2(x) = \sum_{R \in \mathcal{R}} \check{p}_R \hat{V}_2(x, \check{c}_R) \quad \textit{with} \quad \hat{V}_2(x, \check{c}_R) \coloneqq \min_{y \in \mathbb{R}^m} \; \check{c}_R^\top y + \mathbb{I}_{A_2 y + B_2 x \leqslant b}, \tag{2.19}$$

where $\check{p}_R \coloneqq \mathbb{P}\big[\boldsymbol{c} \in \operatorname{ri}(R)\big]$ and $\check{c}_R \coloneqq \mathbb{E}\big[\boldsymbol{c} \,|\, \boldsymbol{c} \in \operatorname{ri}(R)\big]$ if $\check{p}_R > 0$ and $\check{c}_R \coloneqq 0$ if $\check{p}_R = 0$.

In particular, if $\mathcal{R}$ is a refinement of $\bigwedge_{\sigma \in \mathcal{C}(P,\pi)} -\mathcal{N}_\sigma$, Eq. (2.19) holds for all $x \in \pi(P)$.

[1] following traditional two-stage notation, the letter y is used to denote the recourse x_2.

This is a local exact quantization result, since (2.19) shows that $V_2(x)$ coincides with the value function of a second stage problem with a cost distribution supported by the finite set $\{\check{c}_R \mid R \in \mathcal{R}\}$.

Note that $\mathcal{R} = \bigwedge_{\sigma \in \mathcal{C}^{\max}(P,\pi)} -\mathcal{N}_\sigma$ satisfies the condition of Theorem 1 since if τ is a face of σ in the chamber complex, $\mathcal{N}_\sigma$ refines $\mathcal{N}_\tau$ by [RZ96, Lemma 2.2].

Corollary 2.9. *Under assumption 2.1, let $x \in \pi(P)$ and $\sigma \in \mathcal{C}(P,\pi)$ such that $x \in \mathrm{ri}(\sigma)$, then for every refinement $\mathcal{R}$ of $-\mathcal{N}_\sigma$, the subgradient of V at point x is given by the Minkowski sum*

$$\partial V_2(x) = \sum_{R \in \mathcal{R}} \check{p}_R B_2 D(x, \check{c}_R, \tag{2.20}$$

where $D(x,c) \coloneqq \arg\max\{(B_2 x - b_2)^\top \mu : A_2^\top \mu = -c, \mu \geqslant 0\}$, $\check{p}_R \coloneqq \mathbb{P}\big[\boldsymbol{c} \in \mathrm{ri}(R)\big]$ and $\check{c}_R \coloneqq \mathbb{E}\big[\boldsymbol{c} \,|\, \boldsymbol{c} \in \mathrm{ri}(R)\big]$ if $\check{p}_R > 0$ and $\check{c}_R \coloneqq 0$ if $\check{p}_R = 0$.

In particular, if $\mathcal{R}$ is a refinement of $\bigwedge_{\sigma \in \mathcal{C}(P,\pi)} -\mathcal{N}_\sigma$, the subgradient formula Eq. (2.19) *holds for all $x \in \pi(P)$.*

Corollary 2.10 (Exact quantization fo 2-stage stochastic linear problem)**.** *The 2-stage stochastic problem*

$$\min_{x \in \mathbb{R}^n, \boldsymbol{y} \in L_\infty(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{R}^m)} \quad c_1^\top x_1 + \mathbb{E}\big[\boldsymbol{c}_2^\top \boldsymbol{y}\big] \tag{2.21a}$$

$$\text{s.t.} \quad A_1 x + b_1 \leqslant 0, \tag{2.21b}$$

$$A_2 x + B_2 \boldsymbol{y} \leqslant b_2 \quad \text{a.s.}, \tag{2.21c}$$

$$\boldsymbol{y} \preccurlyeq \boldsymbol{c}_2, \tag{2.21d}$$

is equivalent, for every refinement $\mathcal{R}$ of $\bigwedge_{\sigma \in \mathcal{C}(P,\pi)} -\mathcal{N}_\sigma$, to the deterministic linear problem

$$\min_{x \in \mathbb{R}^n, (y_R) \in (\mathbb{R}^m)^{\mathcal{R}}} \quad c^\top x + \sum_{R \in \mathcal{R}} \check{p}_R \check{q}_R^\top y_R \tag{2.22a}$$

$$\text{s.t.} \quad A_1 x + b_1 \leqslant 0, \tag{2.22b}$$

$$A_2 x + B_2 y_R \leqslant b_2\ , \quad \forall R \in \mathcal{R}, \tag{2.22c}$$

where $\check{p}_R \coloneqq \mathbb{P}\big[\boldsymbol{c} \in \mathrm{ri}(R)\big]$ and $\check{q}_R \coloneqq \mathbb{E}\big[\boldsymbol{c} \,|\, \boldsymbol{c} \in \mathrm{ri}(R)\big]$ if $\check{p}_R > 0$ and $\check{q}_R \coloneqq 0$ if $\check{p}_R = 0$.

Note that the chamber complex $\mathcal{R}$ is obtained only from a geometrical analysis of the coupling constraint polyhedron P, we see that the exact quantization methodology presented here is *universal* in the sense that it does not depend on the distribution of $\boldsymbol{c}$. This is exemplified in the following theorem.

Theorem 2 (Universal affine regions)**.** *For all distributions of $\boldsymbol{c}$ satisfying assumption 2.1, the expected cost-to-go function V_2 is affine on each cell of the chamber complex $\mathcal{C}(P,\pi)$.*

More precisely, for all $x \in \pi(P)$,

$$V_2(x) = \max_{\sigma \in \mathcal{C}^{\max}(P,\pi)} \alpha_\sigma^\top x + \beta_\sigma, \tag{2.23}$$

with

$$\begin{cases} \alpha_\sigma &= \sum_{N \in -\mathcal{N}_\sigma} B_2^\top \mu_\sigma(\check{c}_N), \\ \beta_\sigma &= \sum_{N \in -\mathcal{N}_\sigma} -b_2^\top \mu_\sigma(\check{c}_N), \end{cases} \tag{2.24}$$

where $\mu_\sigma(\check{c}_N) \in D(x, \check{c}_N)$ (defined in Corollary 2.9) for $x \in \mathrm{ri}(\sigma)$.

Remark 2.11. *Theorem 2 shows a finite partition such that V_2 is affine on each of its cells. This partition is the coarsest partition satisfying this property. However, for a given cost distribution V_2 can be affine on a coarser partition. Actually, we can exactly describe the affine region of V_2 through the use of* weighted fiber polyhedron*, which is an extension of the fiber polytope of Billera and Sturmfels [BS92]. We refer to [For22, §4.2.2] for the precise results. Finally, in [For22, §4.3.3] we can find an extension of this characterization to the multistage case.*

2.3.2 Propagating chamber complexes through Dynamic Programming

We next show that chamber complexes are propagated through dynamic programming in a way that is universal with respect to the cost distribution. The following Lemma, whose non-trivial proof is illustrated in Fig. 2.3, shows how to obtain (a refinement of) the affine regions of the cost-to-go function V_t. This refinement depends on the affine regions of V_{t+1} and not on the value of V_{t+1}. This enables the construction of a universal exact quantization result, by recursion, for the multistage case.

Lemma 2.12. *Let R be a polyhedral function on $\mathbb{R}^m$ and denote $\mathcal{F}_{\text{low}}\big(\operatorname{epi}(R)\big)$ the set of lower faces of $\operatorname{epi}(R)$. Then, $\mathcal{R} \coloneqq \pi_y^{y,z}\Big(\mathcal{F}_{\text{low}}\big(\operatorname{epi}(R)\big)\Big)$ is a coarsest polyhedral complex such that R is affine on each element of $\mathcal{R}$. Let $\xi = (A, B, b)$ be fixed and assumption 2.1 holds. Define, for all $x \in \mathbb{R}^n$*

$$Q(x, y) \coloneqq R(y) + \mathbb{I}_{Ay+Bx \leqslant b}, \tag{2.25a}$$

$$V(x) \coloneqq \mathbb{E}\big[\min_{y \in \mathbb{R}^m} \boldsymbol{c}^\top y + Q(x, y)\big]. \tag{2.25b}$$

Let $\mathcal{V} \coloneqq \mathcal{C}(\mathcal{F}(P) \wedge (\mathbb{R}^n \times \mathcal{R}), \pi_x^{x,y}) \subset 2^{\mathbb{R}^n}$ with $P \coloneqq \{(x, y) \mid Ay + Bx \leqslant b\}$.
Then, $\mathcal{V} \preccurlyeq \mathcal{C}(\operatorname{epi}(Q), \pi_x^{x,y,z})$ and V is a polyhedral function which is affine on each element of $\mathcal{V}$.

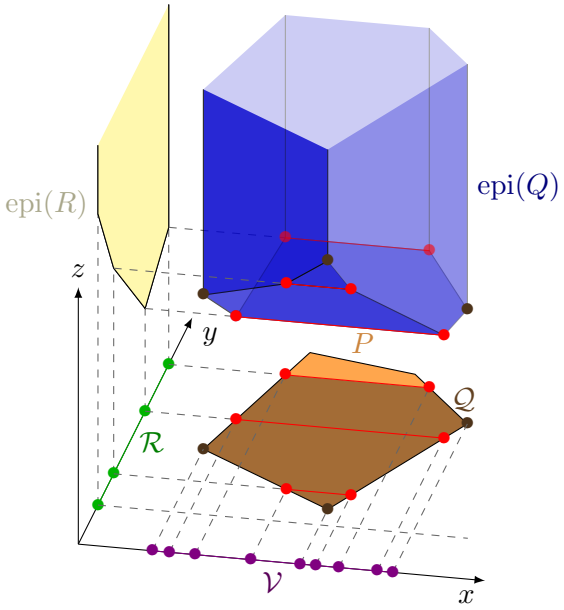


Figure 2.3: An illustration of the proof of Lemma 2.12: the epigraph $\operatorname{epi}(Q)$ of the coupling function in blue in the (x, y, z) space, the epigraph of R in yellow in the (y, z) plane, the affine regions $\mathcal{R}$ of R in green on the y axis, the coupling polyhedron P in orange and brown in the (x, y) plane, the polyhedral complex $\mathcal{Q}$ in red and brown in the (x, y) plane and the chamber complex $\mathcal{V}$ in violet on the x axis.

Remark 2.13. *In Lemma 2.12, the complex $\mathcal{V}$ is independent of the distribution of $\boldsymbol{c}$. However, for special choices of $\boldsymbol{c}$, V might be affine on each cell of a coarser complex than $\mathcal{V}$. For instance, if $R = 0$ and $\boldsymbol{c} \equiv 0$, we have that $V = \mathbb{I}_{\pi_x^{x,y}(P)}$, V is affine on $\pi_x^{x,y}(P)$. Nevertheless, $\mathcal{V} = \mathcal{C}(P, \pi_x^{x,y})$ is generally finer than $\mathcal{F}\big(\pi_x^{x,y}(P)\big)$.*

2.3.3 Exact quantization of MSLP

We next show that the multistage program with arbitrary cost distribution is equivalent to a multistage program with independent, finitely distributed, cost distributions. Further, for all step t, there exist affine regions, independent of the distributions of costs, where V_t is affine. assumption 2.1 is naturally extended to the multistage setting as follows

Assumption 2.2. *The sequence $(\boldsymbol{\xi}_t)_{2\leqslant t\leqslant T}$ is independent.*[1] *Further, for each $t \in [\![2, T]\!]$, $\boldsymbol{\zeta}_t = (\boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)$ is finitely supported, and $\boldsymbol{c}_t \in L^1(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{R}^{n_t})$ is integrable with $\boldsymbol{c}_t \in -\operatorname{cone}(\boldsymbol{A}_t^\top)$ almost surely.*

Note that assumption 2.2 does not require independence between $\boldsymbol{c}_t$ and $\boldsymbol{\zeta}_t$. Let $t \in [T]$. For any $\zeta \coloneqq (A, B, b) \in \operatorname{supp}(\boldsymbol{\zeta}_t)$ we define the coupling polyhedron

$$P_t(\zeta) \coloneqq \{(x_{t-1}, x_t) \in \mathbb{R}^{n_{t-1}} \times \mathbb{R}^{n_t} \mid A_t x_t + B_t x_{t-1} \leqslant b_t\}, \tag{2.26}$$

and consider, for $x_{t-1} \in \mathbb{R}^{n_{t-1}}$,

$$\widetilde{V}_t(x_{t-1}|\zeta) \coloneqq \mathbb{E}\big[\min_{x_t \in \mathbb{R}^{n_t}} \boldsymbol{c}_t^\top x_t + V_{t+1}(x_t) + \mathbb{I}_{A_t x_t + B_t x_{t-1} \leqslant b_t} \mid \boldsymbol{\zeta}_t = \zeta\big]. \tag{2.27}$$

Then, the cost-to-go function V_t is obtained by

$$V_t(x_{t-1}) = \sum_{\zeta \in \operatorname{supp}(\boldsymbol{\zeta}_t)} \mathbb{P}\big[\boldsymbol{\zeta}_t = \zeta\big] \widetilde{V}_t(x_{t-1} \mid \zeta). \tag{2.28}$$

The next two theorems extend the quantization results of Theorem 1 to the multistage settings: an MSLP with non-finitely supported, but independent, random noises is equivalent to an MSLP given on a finite tree.

Theorem 3 (Universal exact quantization, multistage)**.** *Assume that $(\boldsymbol{\xi}_t)_{t\in[T]}$ is a sequence of independent, finitely supported, random variables. We define by induction $\mathcal{P}_{T+1} \coloneqq \{\mathbb{R}^{n_T}\}$ and for $t \in \{2, \dots, T\}$*

$$\mathcal{P}_{t,\zeta} \coloneqq \mathcal{C}(\mathbb{R}^{n_t} \times \mathcal{P}_{t+1} \wedge \mathcal{F}\big(P_t(\zeta)\big), \pi_{x_{t-1}}^{x_{t-1}, x_t}), \tag{2.29a}$$

$$\mathcal{P}_t \coloneqq \bigwedge_{\zeta \in \operatorname{supp}(\boldsymbol{\zeta}_t)} \mathcal{P}_{t,\zeta}. \tag{2.29b}$$

Then, for all costs distributions $(\boldsymbol{c}_t)_{2\leqslant t\leqslant T}$ such that $(\boldsymbol{c}_t, \boldsymbol{\zeta}_t)_{2\leqslant t\leqslant T}$ satisfies assumption 2.2 and all $t \in \{2, \dots, T\}$, we have $\operatorname{supp}(\mathcal{P}_t) = \operatorname{dom}(V_t)$, and V_t is polyhedral and affine on each cell of $\mathcal{P}_t$.

Further, for all $x_{t-1} \in \mathbb{R}^{n_{t-1}}$ and all $\zeta \in \operatorname{supp}(\boldsymbol{\zeta}_t)$, we have a quantized version of Eq. (2.27)*:*

$$\widetilde{V}_t(x_{t-1}|\zeta) = \sum_{N \in \mathcal{N}_{t,\zeta}} \check{p}_{t,N|\zeta} \min_{x_t \in \mathbb{R}^{n_t}} \Big\{\check{c}_{t,N|\zeta}^\top x_t + V_{t+1}(x_t) + \mathbb{I}_{Ax_t + Bx_{t-1} \leqslant b}\Big\}, \tag{2.30}$$

where $\mathcal{N}_{t,\zeta} \coloneqq \bigwedge_{\sigma \in \mathcal{P}_{t,\zeta}} -\mathcal{N}_{t,\zeta,\sigma}$ (with $\mathcal{N}_{t,\zeta,\sigma} \coloneqq \mathcal{N}(\operatorname{epi}(Q_t^\zeta)_{x_{t-1}})$ for an arbitrary $x_{t-1} \in \operatorname{ri}(\sigma)$) and for all $\zeta \in \operatorname{supp}(\boldsymbol{\zeta}_t)$ and $N \in \mathcal{N}_{t,\zeta}$ we denote

$$\check{p}_{t,N|\zeta} \coloneqq \mathbb{P}\big[\boldsymbol{c}_t \in \operatorname{ri} N \mid \boldsymbol{\zeta}_t = \zeta\big], \tag{2.31a}$$

$$\check{c}_{t,N|\zeta} \coloneqq \begin{cases} \mathbb{E}\big[\boldsymbol{c}_t \mid \boldsymbol{c}_t \in \operatorname{ri} N, \boldsymbol{\zeta}_t = \zeta\big] & \text{if } \mathbb{P}\big[\boldsymbol{\zeta}_t = \zeta, \boldsymbol{x} \in \operatorname{ri} N\big] \neq 0 \\ 0 & \text{otherwise} \end{cases}. \tag{2.31b}$$

This theorem then allows to derive complexity results for MSLPs. From Hanasusanto, Kuhn and Wiesemann [HKW16] we have that 2SLP is ♯P-hard, by reducing the computation of the volume of a polytope to the resolution of a 2-stage stochastic program. We show in [VL13] that for a fixed dimension of the recourse space, 2-stage programming is polynomial. We also extend this result to MSLPs. Therefore, the status of 2-stage programming seems somehow comparable to the status of the computation of the volume of a polytope – which is also both ♯P-hard and polynomial when the dimension is fixed (see [Law91]). Another example of ♯P-hard problems that are fixed dimension polynomial is the problem of counting the integer points in a given polytope (see [LJ83]).

More precisely, we show in [§6 VL13] that MSLP with general cost can be solved approximately in polynomial time in $\log(1/\varepsilon)$, when $T, n_2, \cdots, n_T$ are fixed for a large class of regular density functions.

[1] The results can be adapted to non-independent $\boldsymbol{\xi}_t$ as long as $\boldsymbol{c}_t$ is independent of $(\boldsymbol{c}_\tau)_{\tau<t}$ conditionally on $(\boldsymbol{\xi}_{\tau\leqslant t})$.

2.4 Local quantization for stochastic constraints

In Section 2.3 we focused on 2SLP (and MSLPs) with stochastic cost. We now take a dual approach to consider 2SLPs with stochastic constraints. Recall that Example 2.2 proved that we could not obtain a uniform exact quantization in this case, thus we derive a local exact quantization result for 2SLP with generally distributed constraints and fixed recourse.

The main contributions of this section are the following: i) using polyhedral geometry tools we provide a general adapted partition oracle, ii) we give a new necessary and sufficient condition for a partition to be adapted to $\check{x}$ even in the non-finitely supported case, iii) by casting APM methods as accelerated L-Shaped algorithms where tangent cones are added instead of tangent planes (affine cuts), we give convergence and complexity results for APM methods.

Setting, framework and oracle

We consider the following 2-stage stochastic linear problem with fixed recourse:

$$\min_{x\in\mathbb{R}^n_+} \Big\{ c^\top x + \underbrace{\mathbb{E}\big[\hat{V}(x,\boldsymbol{\xi})\big]}_{:=V(x)} \mid Ax = b \Big\}, \tag{2SLP}$$

where the expectation is with respect to $\boldsymbol{\xi} = (\boldsymbol{T}, \boldsymbol{h})$ an integrable random variable on $(\Omega, \mathcal{A}, \mathbb{P})$ taking values in $\Xi \subset \mathbb{R}^{\ell\times n} \times \mathbb{R}^\ell$, and the recourse cost is

$$\hat{V}(x,\xi) \coloneqq \min_{y\in\mathbb{R}^m_+} \Big\{ q^\top y \mid Tx + Wy = h \Big\}. \tag{2.32}$$

The dual formulation of the recourse problem is

$$\hat{V}^D(x,\xi) \coloneqq \max_{\lambda\in\mathbb{R}^\ell} \Big\{ (h - Tx)^\top \lambda \mid W^\top \lambda \leqslant q \Big\}. \tag{2.33}$$

We define

$$X \coloneqq \{x \in \mathbb{R}^n_+ \mid Ax = b\}, \tag{2.34a}$$

$$D \coloneqq \{\lambda \in \mathbb{R}^\ell \mid W^\top \lambda \leqslant q\}. \tag{2.34b}$$

In the rest of the chapter, we assume $D \neq \emptyset$ which implies by duality: $\hat{V}(x,\xi) = \hat{V}^D(x,\xi)$.

For the sake of simplicity, we assume throughout the chapter that we are in a *relatively complete recourse* setting, that is $X \subset \mathrm{dom}(V)$. Most results can be obtained without this assumption if we add feasibility cuts (see § 2.4.2).

Let P be a measurable subset of Ξ. We denote by $\mathbb{E}\big[\,\cdot\,|P\big]$ the conditional expectation $\mathbb{E}\big[\,\cdot\,|\boldsymbol{\xi} \in P\big]$ and $\mathbb{P}\big[P\big]$ the probability $\mathbb{P}\big[\boldsymbol{\xi} \in P\big]$. For technical reasons, we consider that two measurable subsets of $E, F \subset \Xi$ are $\mathbb{P}$-*equivalent*, if and only if they differ by a $\mathbb{P}$-negligeable set, and work with the class of equivalence of these subsets, thus extending the definition of partition to $\mathbb{P}$-partitions, and refinement of partitions. Details can be found in [§2 VL12].

Definition 2.14 (Expected recourse cost of partition)**.** *For $\mathcal{P}$ a $\mathbb{P}$-partition of $\Xi \subset \mathbb{R}^{\ell\times m} \times \mathbb{R}^\ell$ we define*

$$V_{\mathcal{P}} : x \mapsto \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big] \hat{V}\big(x, \mathbb{E}\big[\boldsymbol{\xi}|P\big]\big). \tag{2.35}$$

Let $\check{x} \in \mathrm{dom}(V)$. We say that a $\mathbb{P}$-partition $\mathcal{P}$ is adapted to $\check{x}$ *if*

- *$V_{\mathcal{P}}$ is valid,* i.e., *$V_{\mathcal{P}}(x) = V(x) := \mathbb{E}\big[\hat{V}(x,\boldsymbol{\xi})\big]$ for all $x \in \mathbb{R}^n$.*
- *and $V_{\mathcal{P}}$ is tight at $\check{x}$* i.e., *$V_{\mathcal{P}}(\check{x}) = V(\check{x}) := \mathbb{E}\big[\hat{V}(\check{x},\boldsymbol{\xi})\big]$.*

The following lemma shows that, by convexity, a finer partition yields a larger expected cost-to-go function.

Lemma 2.15. *Let $\mathcal{P}$ and $\mathcal{R}$ two $\mathbb{P}$-partitions of Ξ then*

$$\mathcal{P} \preccurlyeq_{\mathbb{P}} \mathcal{R} \quad \Longrightarrow \quad V_{\mathcal{P}} \geqslant V_{\mathcal{R}}. \tag{2.36}$$

Moreover,

$$V_{\mathcal{P}\wedge\mathcal{R}} \geqslant \max(V_{\mathcal{P}}, V_{\mathcal{R}}). \tag{2.37}$$

Finally,

$$\hat{V}\big(\cdot, \mathbb{E}\big[\boldsymbol{\xi}\big]\big) \leqslant V_{\mathcal{P}} \leqslant V. \tag{2.38}$$

In particular, in this setting with deterministic recourse matrix W and cost q, for all partition $\mathcal{P}$, $V_{\mathcal{P}}$ is valid. We then only have to prove $V_{\mathcal{P}}(\check{x}) = V(\check{x})$ to prove that $\mathcal{P}$ is adapted to $\check{x}$. However, this would not be the case when we extend to general cost $\boldsymbol{q}$.

With those definitions, we present in Algorithm 3 a generic framework for APM methods.

$k \leftarrow 0$, $z_U^0 \leftarrow +\infty$, $z_L^0 \leftarrow -\infty$, $\mathcal{P}^0 \leftarrow \{\Xi\}$;
while $z_U^k - z_L^k > \varepsilon$ **do**
  $k \leftarrow k + 1$;
  Solve $z_L^k \leftarrow \min_{x\in X} c^\top x + V_{\mathcal{P}^{k-1}}(x)$ and let x^k be an optimal solution ;
  Call the oracle on x^k yielding $\mathcal{P}_{x^k}$;
  $\mathcal{P}^k \leftarrow \mathcal{P}^{k-1} \wedge \mathcal{P}_{x^k}$;
  $z_U^k \leftarrow \min\left(z_U^{k-1}, c^\top x^k + V_{\mathcal{P}^k}(x^k)\right)$;

Algorithm 3: Generic framework for APM.

2.4.1 Coarsest adapted partition

In this section, we define $\mathcal{R}_{\check{x}}$, a particular $\mathbb{P}$-partition, and prove that it is, in a generic case, the coarsest partition adapted to $\check{x} \in X$, *i.e.*, the only partition adapted to $\check{x}$ that refines $\mathcal{R}_{\check{x}}$ is $\mathcal{R}_{\check{x}}$ itself. Indeed, we are looking for partitions that yield a precise approximation of recourse cost (exact at $\check{x}$ in the adapted case), while having the smallest possible number of elements.

When the distributions have finite support, [SL15] characterized the partitions adapted to $\check{x}$. Building on this result, a sufficient condition for continuous distribution can be found in [RPM21, Prop. 2]. We now prove that, for any distribution, a partition is adapted to $\check{x}$ if and only if it refines the collection $\overline{\mathcal{R}}_{\check{x}}$ defined in (2.40b). Unfortunately, $\overline{\mathcal{R}}_{\check{x}}$ is not necessarily a $\mathbb{P}$-partition, thus we also provide a partition $\mathcal{R}_{\check{x}} \preccurlyeq \overline{\mathcal{R}}_{\check{x}}$ (see Figure 2.4 for an illustration).

Recall that $D = \{\lambda \in \mathbb{R}^\ell \mid W^\top\lambda \leqslant q\}$ and that the normal cone of D at λ is the set $N_D(\lambda) \coloneqq \{\psi \in \mathbb{R}^\ell \,|\, \psi^\top(\lambda' - \lambda) \leqslant 0, \forall \lambda' \in D\}$. We denote by $\mathrm{ri}(N)$ the relative interior of a cone N. Let $\mathcal{N}(D) \coloneqq \{N_D(\lambda) \,|\, \lambda \in D\}$ be the normal fan of D, *i.e.,* the (finite) collection of all normal cones of D. We denote by $\mathcal{N}(D)^{\max} \coloneqq \{N \in \mathcal{N}(D) \,|\, \forall N' \in \mathcal{N}(D), N \subset N' \Rightarrow N = N'\}$ the collection of the maximal elements of $\mathcal{N}(D)$ (*i.e.,* full dimensional cones up-to lineality spaces).

Theorem 4. *Fix $\check{x} \in \mathrm{dom}(V)$ and N a cone in $\mathbb{R}^m$. We define $E_{N,\check{x}}$ and $\overline{E}_{N,\check{x}}$, subsets of Ξ, as*

$$E_{N,\check{x}} \coloneqq \{\xi \in \Xi \mid h - T\check{x} \in \mathrm{ri}(N)\}, \tag{2.39a}$$

$$\overline{E}_{N,\check{x}} \coloneqq \{\xi \in \Xi \mid h - T\check{x} \in N\}. \tag{2.39b}$$

We define $\mathcal{R}_{\check{x}}$ and $\overline{\mathcal{R}}_{\check{x}}$ as

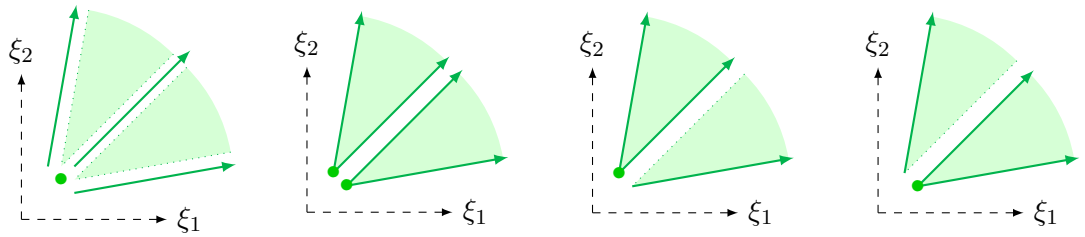
$$\mathcal{R}_{\check{x}} \coloneqq \big\{E_{N,\check{x}} \mid N \in \mathcal{N}(D)\big\}, \tag{2.40a}$$

$$\overline{\mathcal{R}}_{\check{x}} \coloneqq \big\{\overline{E}_{N,\check{x}} \mid N \in \mathcal{N}(D)^{\max}\big\}. \tag{2.40b}$$

Then,

$$\mathcal{P} \preccurlyeq_{\mathbb{P}} \mathcal{R}_{\check{x}} \implies V_{\mathcal{P}}(\check{x}) = V(\check{x}), \tag{2.41a}$$

$$\mathcal{P} \preccurlyeq_{\mathbb{P}} \overline{\mathcal{R}}_{\check{x}} \iff V_{\mathcal{P}}(\check{x}) = V(\check{x}). \tag{2.41b}$$



(a) $\mathcal{R}_{\check{x}}$ (b) $\overline{\mathcal{R}}_{\check{x}}$ (c) $\mathcal{P}$ (d) $\mathcal{P}'$

Figure 2.4: $\mathcal{R}_{\check{x}}$ is a partition of Ξ into 6 elements, $\overline{\mathcal{R}}_{\check{x}}$ is not a partition, $\mathcal{P}$ and $\mathcal{P}'$ are two distinct coarsest partitions (into 2 elements) with $\mathcal{R}_{\check{x}} \preccurlyeq \mathcal{P} \preccurlyeq \overline{\mathcal{R}}_{\check{x}}$ and $\mathcal{R}_{\check{x}} \preccurlyeq \mathcal{P}' \preccurlyeq \overline{\mathcal{R}}_{\check{x}}$.

Remark 2.16. *When the distribution of $\boldsymbol{\xi}$ is absolutely continuous with respect to the Lebesgue measure of Ξ, $\mathcal{R}_{\check{x}} \sim_{\mathbb{P}} \overline{\mathcal{R}}_{\check{x}}$, thus $\mathcal{R}_{\check{x}}$ is the coarsest partition adapted to $\check{x} \in \mathrm{dom}(V)$.*

If $\boldsymbol{\xi}$ does not admit a density, $\mathcal{R}_{\check{x}}$ is still an adapted partition but not necessarily the coarsest, which might not exist (see Fig. 2.4). Nevertheless, any adapted partition should refine $\overline{\mathcal{R}}_{\check{x}}$. Unfortunately, we cannot use $\overline{\mathcal{R}}_{\check{x}}$ in Algorithm 3, as we cannot guarantee that $\overline{\mathcal{R}}_{\check{x}}$ is a $\mathbb{P}$-partition.

Remark 2.17. *Note that Proposition 2 of [RPM21] implies that all partition oracle returning partitions satisfying assumption (7) of [RPM21] must be refinements of $\overline{\mathcal{R}}_{\check{x}}$ by Theorem 4. In the finite scenario case, our adaptedness condition is equivalent to Song and Luedtke's condition [SL15].*

The proof of Theorem 4 relies on the following technical lemma which underlines the difference between $\mathcal{R}_{\check{x}}$ and $\overline{\mathcal{R}}_{\check{x}}$.

Lemma 2.18. *Consider a set $P \subset \Xi$ such that $\mathbb{P}(P) > 0$, and a first-stage control $\check{x} \in \mathrm{dom}(V)$. Then,*

$$\exists R \in \mathcal{R}_{\check{x}}, \quad P \subset_{\mathbb{P}} R \tag{2.42}$$

$$\implies \hat{V}(\check{x}, \mathbb{E}\big[\boldsymbol{\xi}|P\big]) = \mathbb{E}\big[\hat{V}(\check{x}, \boldsymbol{\xi})|P\big], \tag{2.43}$$

$$\exists \overline{R} \in \overline{\mathcal{R}}_{\check{x}}, \quad P \subset_{\mathbb{P}} \overline{R} \tag{2.44}$$

$$\iff \hat{V}(\check{x}, \mathbb{E}\big[\boldsymbol{\xi}|P\big]) = \mathbb{E}\big[\hat{V}(\check{x}, \boldsymbol{\xi})|P\big]. \tag{2.45}$$

2.4.2 Comparison with other algorithms and convergence

We show that the partition-based methods can be seen as an acceleration of the cutting plane method, which yields a finite convergence proof with a bound on the number of steps.

The following lemma shows that for any first-stage control $x \in X$, if the partition is adapted to x, then the subdifferential of approximate expected recourse cost coincides with the subdifferential of the true expected recourse cost.

Lemma 2.19. *Let $\check{x} \in \mathrm{dom}(V)$ and $\mathcal{P}$ be a refinement of $\mathcal{R}_{\check{x}}$, i.e. $\mathcal{P} \preccurlyeq \mathcal{R}_{\check{x}}$, then*

$$\partial V_{\mathcal{R}_{\check{x}}}(\check{x}) \subset \partial V_{\mathcal{P}}(\check{x}) \subset \partial V(\check{x}). \tag{2.46}$$

Furthermore, if $\check{x} \in \mathrm{ri}(\mathrm{dom}(V))$*,*

$$\partial V_{\mathcal{R}_{\check{x}}}(\check{x}) = \partial V_{\mathcal{P}}(\check{x}) = \partial V(\check{x}). \tag{2.47}$$

The classical L-shaped method (see e.g. [BL11, Chapter 5]) is a specification of Benders decomposition to 2SLP with finitely supported distributions. The core idea consists in representing the expected recourse cost in (2SLP), by a lift variable

$$\min_{x\in X, \theta\in\mathbb{R}} \Big\{ c^\top x + \theta \mid (x,\theta) \in \mathrm{epi}(V) \Big\}. \tag{2.48}$$

We then relax the epigraphical representation $(x,\theta) \in \mathrm{epi}(V)$, replacing it by a set of valid inequalities called *cuts*, i.e.

$$\begin{aligned} \min_{x\in X, \theta\in\mathbb{R}} \quad & c^\top x + \theta && && \text{(2.49a)}\\ \text{s.t.} \quad & g^\top x + v \leqslant \theta, && \forall (g,v) \in \mathcal{O}, && \text{(2.49b)}\\ & f^\top x \leqslant \overline{f}, && \forall (f,\overline{f}) \in \mathcal{F}. && \text{(2.49c)} \end{aligned}$$

More precisely, assume that we have such a relaxation of (2SLP). Let x^k be an optimal first-stage control of this relaxation. If it is admissible, meaning that for all scenario ξ there exists an admissible recourse control y_ξ, we compute, through duality, a subgradient $g^k \in \partial V(x^k)$. This yields a new *optimality cut* $\theta \geqslant (g^k)^\top (x - x^k) + V(x^k)$, which is added to $\mathcal{O}$. If x^k is not admissible we can add a *feasibility cut* to $\mathcal{F}$ instead by using dual optimal extreme ray (see [BL11, §5.1.b]). We then solve our strengthened relaxation to obtain x^{k+1}.

The L-Shaped method specifies that the subgradient g^k can be obtained as an average over ξ of subgradients $g^{k,\xi} \in \partial_x Q(x^k, \xi)$. In particular, it means that, to compute the subgradient, we can solve $|\mathrm{supp}(\boldsymbol{\xi})|$ smaller LP instead of a large one.

Remark 2.20 (L-shaped for continuous distribution)**.** *When the distributions are non-finitely supported, we cannot apply naively this method as there is a non-finite number of scenarios. Nevertheless, we can still approximate* $\mathrm{epi}(V)$ *with cuts. We can compute* $\theta = V_{\mathcal{R}_{\check{x}}}(\check{x})$ *and a subgradient* $g \in \partial V_{\mathcal{R}_{\check{x}}}(\check{x})$ *by solving* $|\mathcal{R}_{\check{x}}|$ *linear problems of the form* (2.33) *through exact quantization. By Theorem 4,* $\theta = V_{\mathcal{R}_{\check{x}}}(\check{x}) = V(\check{x})$*. Further,* $g \in \partial V_{\mathcal{R}_{\check{x}}}(\check{x}) \subset \partial V(\check{x})$ *by Lemma 2.19. Then* (θ, g) *defines an optimality cut.*

Lemma 2.19 shows that at each step k of Algorithm 3, we add a collection of valid cuts which are exact at x^k to our collection of cuts. This means that APM methods can be seen as a Bender's decomposition method where we add more than one exact cut per iteration. In particular, when $x^k \in \mathrm{ri}\big(\mathrm{dom}(V)\big)$ we add the whole tangent cone of $\mathrm{epi}(V)$ at x instead of a single cut. Thus, as the bounds generated by Algorithm 3 are monotonic, we can adapt the classical proof of Kelley's cutting plane algorithm to APM and derive the following convergence theorem.

Theorem 5. *Assume that the partition oracle used is adapted. If* $X \subset \mathbb{R}^n_+$ *has a finite diameter* $M \in \mathbb{R}_+$ *and* $x \mapsto c^\top x + V(x)$ *is Lipschitz with constant* L *then the partition-based Algorithm 3 finds an* ε*-solution in at most* $\big(\frac{LM}{\varepsilon} + 1\big)^n$ *iterations.*

Remark 2.21. *The results presented here are given for deterministic cost* q*. By taking the common refinement we can easily extend them to the finitely supported* $\boldsymbol{q}$ *case. Perhaps more surprisingly we can also extend the result to any distribution of* $\boldsymbol{q}$*. Indeed, we can exactly quantize the cost* $\boldsymbol{q}$ *through the collection of relative interiors of the secondary cones of* W*. This theoretical, and quite technical, derivation can be found in [§2 VL16].*

Chapter 3

Exact lower bounds in multistage stochastic problem

In this chapter we are going to discuss a class of algorithms for MSP, satisfying Assumption (SWI), called *Trajectory Following Dynamic Programming* (TFDP) algorithms, which encompass the well-known Stochastic Dual Dynamic Programming (SDDP) algorithm and numerous variants.

This class of algorithm works by constructing lower bounds of the value functions, and sometimes also maintaining upper bounds. The framework presented here was developed for [VL16] building on contributions made in [VL3] and [VL10]. The convergence proof given here follows [VL16], as the results are more precise than in previous contributions, and do not require the finitely supported noise Assumption (FSN). Construction of upper bounds is discussed in Chapter 4.

We first present in Section 3.1 the algorithmic framework, various required assumptions, and discuss variants. Then, in Section 3.2 we provide convergence results for this class of algorithm.

3.1 Trajectory Following Dynamic Programming algorithms framework

For ease of reference, we recall the risk-neutral multistage stochastic Problem (1.1)

$$\begin{aligned} \underset{(\boldsymbol{x}_t)_{t\in[T]}\in L^0(\Omega,\mathcal{A},\mathbb{P};\mathbb{R}^{nT})}{\operatorname{Min}} \quad & \mathbb{E}_{\mathbb{P}}\left[\sum_{t=1}^{T} c_t(\boldsymbol{x}_{t-1},\boldsymbol{x}_t,\boldsymbol{\xi}_t) + V_{T+1}(\boldsymbol{x}_T)\right] && \text{(MSP)}\\ \text{s.t.} \quad & (\boldsymbol{x}_{t-1},\boldsymbol{x}_t) \in P_t(\boldsymbol{\xi}_t) && \forall t\in[T],\\ & \boldsymbol{x}_t \preccurlyeq \mathcal{A}_t \coloneqq \sigma(\boldsymbol{\xi}_{[t]}) && \forall t\in[T]. \end{aligned}$$

With stagewise independent noise assumption, we recall the Bellman equation (1.20)

$$V_t = \mathcal{B}_t(V_{t+1}), \qquad \forall t\in[T],$$

where V_t is the expected, or more generally risk adjusted, cost-to-go function defined in Eq. (1.19).

The main idea of TFDP algorithms consists in iteratively refining lower (and sometimes upper) approximations of the risk-adjusted cost-to-go functions V_t. More precisely, at each iteration, we determine, in a forward phase, trial points at which the approximations should be refined. Then, in a backward phase, we construct *cuts*, which are functions that under-approximate V_t. These cuts are as close as possible to the true risk-adjusted cost-to-go functions around the trial points. The lower approximations are finally defined as the maximum of computed cuts.

3.1.1 Assumptions

We describe here the required assumption to prove the convergence of a TFDP algorithm. First, we make assumptions over the MSP problem, and then over elements of the TFDP algorithm.

Setting assumption

Using notations from Chapter 1, we make the following assumptions.

Assumption 3.1 (Compatibility of constraints)**.** *We make the following assumptions, for all $t \in [T]$,*

i) c_t *is a proper normal integrand;*

ii) for all $(x_{t-1} \in X_{t-1}^r)$ *(see* (1.14)*), and all* $(\boldsymbol{x}_t \in \mathcal{X}_t(x_{t-1}, \boldsymbol{\xi}_t))$ *the random variable* $c_t(x_{t-1}, \boldsymbol{x}_t, \boldsymbol{\xi}_t)$ *is integrable (in particular* $c_t(x_{t-1}, x_t, \boldsymbol{\xi}_t) < +\infty$ $\mathbb{P}$*-almost surely);*

iii) for all $x_{t-1} \in X_{t-1}^r$ *and almost all* $\xi_t \in \Xi_{t-1}$*,* $\mathcal{X}_t(x_{t-1}, \xi_t)$ *(see* (1.13)*) is a non-empty compact subset of* $\mathbb{R}^{n_t}$*.*

Assumption 3.1 ensures that (MSP) is well-posed and admits an optimal solution. It also guarantees that we are in a *relatively complete recourse* setting (see Definition 1.19) in the sense that any sequence of variable $(\boldsymbol{x}_\tau)_{\tau \leqslant t}$ satisfying $\boldsymbol{x}_\tau \in X_\tau(\boldsymbol{x}_{\tau-1}, \boldsymbol{\xi}_\tau)$, for $\tau \leqslant t$ can be completed into an admissible policy $(\boldsymbol{x}_\tau)_{\tau \leqslant T}$ such that $\mathbb{E}\big[\sum_{t=1}^T c_t(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t, \boldsymbol{\xi}_t)\big] < +\infty$.

As we are considering Dynamic Programming methods, the stagewise independence Assumption (SWI) is assumed to hold true.

We have a last non-trivial assumption over the problem setting.

Assumption 3.2 (Lipschitz)**.** *For* $t \in [T]$*, we assume that*[1]

i) X_t^r *has a diameter smaller than* $D_t < +\infty$*;*

ii) the expected cost-to-go function V_t *is* L_t*-Lipschitz.*

Both parts of assumption 3.2 are strong requirements, needed for the convergence results, while still being natural in most settings. Part i) is satisfied for example if assumption 3.1 holds, $\mathcal{X}_t(x_{t-1}, \cdot)$ is Lipschitz for all $x_{t-1} \in X_{t-1}^r$ and all Ξ_t are bounded. Part ii) is satisfied under assumption 3.1 in the linear case, or through an *extended relatively complete recourse* assumption, in the convex case (see [VL3]), which requires that state x_t that are slightly outside X_t^r are still admissible.

Requirement of TFDP algorithms

As said above, TFDP algorithms maintain approximations of the risk-adjusted cost-to-go. These approximations need to be *admissible* as defined next.

Assumption 3.3 (Admissible approximations)**.** *The computed cuts* f_t^k *of* $\mathcal{B}_t(\underline{V}_{t+1}^k)$ *at* x_t^k *satisfy:*

i) f_t^k *is* $\underline{\gamma}_t$*-tight, i.e.* $f_t^k(x_t^k) \geqslant \mathcal{B}_t(\underline{V}_{t+1}^k)(x_t^k) - \underline{\gamma}_t$

ii) f_t^k *is valid, i.e.* $f_t^k \leqslant \mathcal{B}_t(\underline{V}_{t+1}^k)$

iii) $\overline{V}_t^k$ *is* $\overline{L}_t$*-Lipschitz*

On the other hand, the upper approximation $\overline{V}_t^k$*, not necessarily computed, shall satisfy the following properties:*

iv) $\overline{V}_t^k(x_t^k) \leqslant \mathcal{B}_t(\overline{V}_{t+1}^k)(x_t^k) + \overline{\gamma}_t$ *(tightness)*

v) $\overline{V}_t^k \geqslant \mathcal{B}_t(\overline{V}_{t+1}^k)$ *(validity)*

vi) $\overline{V}_t^k \leqslant \overline{V}_t^{k-1}$ *(monotonicity)*

[1] We do not necessarily require the knowledge of the diameters or Lipschitz constants.

vii) $\overline{V}_t^k$ *is* $\overline{L}_t$*-Lipschitz*

For the algorithm to be well-defined we need to guarantee the existence of cuts and upper approximation satisfying previous assumptions, as formally assumed now:

Assumption 3.4. *For every* $t \in [T]$ *and* $k \in \mathbb{N}^\star$*, there exists at least one cut* f_t^k *of* $\mathcal{B}_t(\underline{V}_{t+1}^k)$ *satisfying assumption 3.3.*

To prove convergence we finally need some assumption over the node selection process that is discussed in Section 3.2, satisfied for example if the nodes are selected randomly as usual in SDDP.

3.1.2 Algorithm framework

To define a TFDP algorithm, we need to define a *node selection* procedure, a forward Bellman operator, a cut computation procedure and an upper bound update. With these elements defined, the algorithm goes iteratively as follows: in a forward phase, we construct a trial state trajectory along the node selected by the node selection process using the forward Bellman operator. Then, in a backward phase, we update the lower approximation around the trial point (and the upper approximation if needed). This is made formal in Algorithm 4.

Data: Random variables $\boldsymbol{\xi}_t$, cost function at each step c_t, constraints set-valued function X_t, initial state x_0, γ_t^F-forward operators $\mathcal{F}_t$.

1 $\underline{V}_t^0 \equiv -\infty$ and $\overline{V}_t^0 \equiv +\infty$ for $t \in [T]$;

2 **for** $k \in \mathbb{N}$ **do**

`/* Forward phase */`

3 Set $x_0^k = x_0$;

4 **for** $t = 1 : T-1$ **do**

5 Choose $\xi_t^k \in \mathrm{supp}(\boldsymbol{\xi}_t)$; `// node selection`

6 Let $x_t^k = \mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \xi_t^k)$;

`/* Backward phase */`

7 Set $\underline{V}_T^k \equiv \overline{V}_T^k \equiv 0$;

8 **for** $t = T-1 : -1 : 1$ **do**

9 Find a $\underline{L}_t$-Lipschitz on X_t^r, valid and $\underline{\gamma}$-tight cut f_t^k of $\mathcal{B}_t(\underline{V}_{t+1}^k)$ at x_t^k, i.e. such that $f_t^k(x_t^k) \geqslant \mathcal{B}_t(\underline{V}_{t+1}^k)(x_t^k) - \underline{\gamma}_t$ and $f_t^k \leqslant \mathcal{B}_t(\underline{V}_{t+1}^k)$;

10 Set $\underline{V}_t^k = \max(\underline{V}_t^{k-1}, f_t^k)$;

11 Define $\overline{V}_t^k$ satisfying assumption 3.3, Items iv) to vii) ;

Algorithm 4: A general framework for TFDP algorithms

Remark 3.1 (Asymmetry of upper and lower approximations)**.** *The framework is not symmetrical in its treatment of the upper and lower cost-to-go approximations. Indeed, Line 6 should not be done with the upper approximations*[1] *as it would restrict the exploration of the state space. For example, assume that* $\overline{V}_t$ *are (slightly Lipschitz-regularized) indicator functions of a single point, then the forward phase would always produce the same trajectory, and the upper bound would not be updated.*

Further, multiple TFDP algorithms do not actually compute $\overline{V}_t$*, simply setting it to the true expected cost-to-go* V_t *(for iterations bounds).*

Remark 3.2 (The standard SDDP algorithm)**.** *The most common TFDP algorithm is the stochastic dual dynamic programming (SDDP). It was originally designed by Pinto and Pereira ([PP91]) for multistage stochastic linear problems. In SDDP, the value of the noise* ξ_t^k*, chosen in Line 5, is drawn randomly on* $\mathrm{supp}(\boldsymbol{\xi}_t)$ *which is assumed to be finite. The lower approximations are defined as the maximum of affine cuts. For each* $\xi \in \mathrm{supp}(\boldsymbol{\xi}_t)$*, computing* $\hat{\mathcal{B}}_t(\underline{V}_{t+1}^{k-1})(x_t^k, \xi)$ *consists in solving an LP,*

[1] The upper approximations $(\overline{V}_t^k)_{t\in[T]}$ still provide an admissible policy through the forward Bellman operators which has interesting properties, see Section 4.3, or [VL10].

and standard linear programming duality yields a subgradient $\hat{\alpha}_t^k(\xi) \in \partial\hat{\mathcal{B}}_t(\underline{V}_{t+1}^{k-1})(x_t^k, \xi)$ *and value* $\hat{\beta}_t^k(\xi) = \hat{\mathcal{B}}_t(\underline{V}_{t+1}^{k-1})(x_t^k, \xi)$. *Taking the expectation, we set* $\alpha_t^k = \mathbb{E}\big[\hat{\alpha}_t^k(\boldsymbol{\xi}_t)\big]$ *and* $\beta_t^k = \mathbb{E}\big[\hat{\beta}_t^k(\boldsymbol{\xi}_t)\big]$, *to define the so-called Benders' cut*

$$f_t^k : x_t \mapsto \alpha_t^{k\top}(x_t - x_t^k) + \beta_t^k. \tag{3.1}$$

Under relatively complete recourse assumption, the cuts can be assumed to be $\underline{L}_t$*-Lipschitz. Further, in this simple setting, all errors are null:* $\underline{\gamma}_t = \overline{\gamma}_t = \gamma_t^F = 0$. *Note that no upper bounds are computed and the complexity results of Section 3.2 are obtained by taking* $\overline{V}_t^k = V_t^k$.

Algorithm 4 is a flexible framework, and some lines remain to be detailed, which we now discuss.

Node selection choice in Line 5 Most TFDP algorithms choose ξ_t^k by drawing it randomly according to the law of the random variable $\boldsymbol{\xi}_t^k$. The forward phase can then be seen as a Monte Carlo method for finding a trajectory $x_{[T]}^k$. Then, it is also possible to choose ξ_t^k thanks to quasi-Monte Carlo methods.

Another way of choosing ξ_t^k consists in picking the $\xi \in \operatorname{supp}(\boldsymbol{\xi}_t)$ that maximizes a certain criterion. In [BDZ17], Baucke, Downward and Zakeri suggested to chose ξ_t^k such that x_t^k maximizes the gap between the upper and lower approximations, *i.e.,* $\overline{V}_t^k(x_t^k) - \underline{V}_t^k(x_t^k)$. They called this choice of ξ_t^k, the *problem child* node selection. In [Lan20], Lan presented the Explorative Dual Dynamic Programming algorithm, where ξ_t^k is chosen so that x_t^k is the most distinguishable point, i.e. such that x_t^k is far from the previously computed points, we speak of *explorative* node selection.

The proofs of convergence are harder to derive when ξ_t^k is chosen randomly, and the best upper bound known on the number of iterations of these algorithms are exponential in the horizon T. In comparison, when ξ_t^k is chosen deterministically as the problem child or as the most distinguishable point, the number of iterations is bounded by a polynomial in T. However, random sampling is often more efficient in practice (and easier to implement).

Forward operator choice in Line 6 In most algorithms, we assume that $\gamma_t^F = 0$ for all $t \in [T-1]$, thus $\mathcal{F}_{t-1}(\widetilde{V})(x, \cdot)$ is a measurable selection of $\arg\min_{y \in \mathcal{X}_t(x,\cdot)} c_t(y, \cdot) + \widetilde{V}(y)$. There has also been propositions to use inexact cuts [Gui20] to alleviate the computational burden of each iteration.

Further, there have been various propositions to regularize the SDDP algorithm, see [AP18, VAdOS19, GLT20]. They mostly boil down to choosing a different forward operator, *e.g.,* by adding a regularization term, which can be seen as γ_t^F-forward operator with $\gamma_t^F \neq 0$.

Finally, it is important that the algorithm use a single γ_t^F-forward operator. Indeed, if the set of γ_t^F-optimal solutions $\mathcal{X}^\sharp_{\gamma_t^F, t}(\widetilde{V})(x, \xi)$ is not reduced to a single point, the convergence results only hold for the points selected by the forward operator. This remark is not only theoretical and has implications in practice[1]: to be safe one should use the same solver (and parameters) during the training phase and exploitation phase of the algorithm.

Cuts f_t^k choice in Line 9 We need to compute cuts f_t^k to approximate $\mathcal{B}_t(\underline{V}_{t+1}^k)$ in the neighborhood of x_{t-1}^k. Recall that in Eq. (1.19b), $\mathcal{B}_t$ is defined as an expectation of parametric Bellman operators $\mathcal{B}_t(\underline{V}_{t+1}^k) = \mathbb{E}\big[\hat{\mathcal{B}}_t(\underline{V}_{t+1}^k)(\cdot, \boldsymbol{\xi}_t)\big]$ Eq. (4.1a). Then, we can compute the average cut f_t^k thanks to parametric cuts $\hat{f}_{t,\xi}^k$. In the finitely supported case, we can compute the average cut f_t^k directly by taking $f_t^k = \mathbb{E}\big[\hat{f}_{t,\boldsymbol{\xi}}^k\big]$ whereas in the convex, non-finitely-supported case, there exists methods to approximate $\mathbb{E}\big[f_{t,\boldsymbol{\xi}}^k\big]$. Finally, exact methods for linear problems are developed in [VL16]. Furthermore, depending on the problem structure, there exist several types of parametrized cuts $\hat{f}_{t,\xi}^k$ in the literature. We recall them in [VL16].

[1] For example, consider a problem with two equivalent storage and that only one of them is required to provide an optimal solution. Consider two forward operators, the first one, $\mathcal{F}_{t-1}^1$, prefers using the first storage while the second, $\mathcal{F}_{t-1}^2$ prefers using the second storage. Now assume that the algorithm ran until convergence with $\mathcal{F}_{t-1}^1$ yielding the approximations $\underline{V}_t^\infty$. Then, $\underline{V}_t^\infty$ correctly evaluates the value of the first storage, but has no information on the second. Consequently, a trajectory given by $\mathcal{F}_{t-1}^2(\underline{V}_t^\infty)$ might be far from optimal.

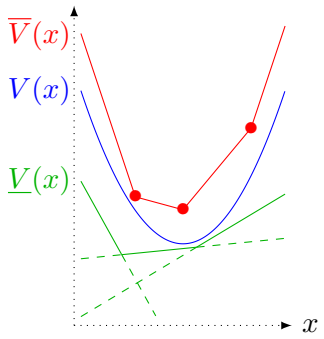


Figure 3.1: An example of upper and lower approximations

Upper approximations $\overline{V}_t^k$ choice in Line 11 In most TFDP algorithms, no upper bound function is computed. In that case, we just set $\overline{V}_t^k \equiv V_t$ in the convergence proof. However, some algorithms rely on the computation of these upper bounds, for example for computing a problem-child node selection. Chapter 4 is dedicated to computing upper bound.

3.1.3 Extensions of the framework

Although we tried to present a general framework, for the sake of simplicity, Algorithm 4 does not integrate every variant of SDDP. We now discuss how this framework can be extended and if the complexity results and proofs are still valid with these new extensions.

Multiple forward phases. In practice, SDDP is often implemented with multiple forward phases, *i.e.,* at iteration k we compute N forward phases $(x_t^{k,i})_{t\in[T-1],i\in[N]}$, in parallel. Consequently, in the backward phase we compute, for each time step $t \in [T-1]$, N tight and valid cuts $(f_t^{k,i})_{i\in[N]}$. This variation is included in the framework of Algorithm 4 by considering that the cut f_t^k is the maximum over $i \in [N]$ of all cuts $f_t^{k,i}$. The complexity results follow directly (in iteration number).

Multicut. In the finitely supported case, instead of computing an average cut f_t^k of the expected cost-to-go function V_t, it is possible to store for each $\xi \in \mathrm{supp}(\boldsymbol{\xi}_t)$ a cut $\hat{f}_{t,\xi}$ of the cost-to-go function $\hat{V}_t(\cdot,\xi)$. Unlike the single-cut case where $\underline{V}_t^k(\cdot) = \max_{\kappa\leqslant k} f_t^\kappa(\cdot)$, in the multicut case, we compute approximation function as $\underline{V}_t^k(\cdot) = \sum_{\xi\in\mathrm{supp}(\boldsymbol{\xi}_t)} \mathbb{P}[\xi] \max_{\kappa\leqslant k} \hat{f}_{t,\xi}^\kappa(\cdot)$. Up to a slight reinterpretation, by considering a global cut $f_t(\cdot) = \sum_{\xi\in\mathrm{supp}(\boldsymbol{\xi}_t)} \mathbb{P}[\xi] \max_{\kappa\leqslant k} \hat{f}_{t,\xi}(\cdot)$, this variation is covered by our framework.

However, with continuous random variables, the notion of multiple cuts is not well-defined.

Cut computation in forward. Another variation of SDDP consists in computing the cuts during the forward phase (and no backward phase). In this variant, the cuts do not approximate $\mathcal{B}_t(\underline{V}_{t+1}^k)$ and $\mathcal{B}_t(\overline{V}_{t+1}^k)$ in the neighborhood of x_t^k, but approximate $\mathcal{B}_t(\underline{V}_{t+1}^{k-1})$ and $\mathcal{B}_t(\overline{V}_{t+1}^{k-1})$ in the same neighborhood. Although this variant is not handled by the framework, all proofs can be adapted straightforwardly.

Cut selection. After many iterations, the number of cuts can slow down the new iterations. To speed up SDDP iterations, another idea is to delete some cuts. For example, we can decide to delete only the dominated cuts, *i.e.,* the cuts that do not affect the values of the approximations $\underline{V}_t^k$. The monotonicity property and the complexity results are still valid in this setting. Unfortunately, finding which cut is dominated is time-consuming which does not make this method numerically efficient. Instead, we often use some heuristics to delete cuts that are probably dominated. However, these heuristics do not guarantee that we have the monotonicity property of approximations. Then, the complexity and convergence results seem harder to obtain.

Adaptive partition-based methods In [SL15], Song and Luedtke presented the adaptive partition-based methods (APM) to solve 2-stage linear problems by partitioning the set of scenarios. It was then adapted to the multistage case in [SS22] where Siddig and Song proposed an adaptive partition-based SDDP, in both case under the finitely supported noise Assumption (FSN). The idea of APM is to replace the expected cost-go-function $V = \mathbb{E}\big[\hat{V}(\cdot, \boldsymbol{\xi})\big]$ by a partitioned expected cost-to-go function $V_{\mathcal{P}} = \sum_{P\in\mathcal{P}} \mathbb{P}\big[\boldsymbol{\xi} \in P\big]\hat{V}(\cdot, \mathbb{E}\big[\boldsymbol{\xi} \,|\, \boldsymbol{\xi} \in P\big])$ where $\mathcal{P}$ is a partition of the uncertainty set Ξ. A partition $\mathcal{P}$ is said to be *tight at* $\check{x}$, if $V_{\mathcal{P}}(\check{x}) = V(\check{x})$, *valid* if $V_{\mathcal{P}}(x) \leqslant V(x)$ for all $x \in \mathbb{R}^{n_t}$ and *adapted to* $\check{x}$ if it is valid and tight at $\check{x}$ (see § 2.4.1). Then, when $\mathcal{P}$ is a partition adapted to $\check{x}$, we can see the partitioned expected cost-to-go function $V_{\mathcal{P}}$ as a valid and tight cut of V at $\check{x}$. Such cuts represent the tangent cone of $\mathrm{epi}(\mathcal{B}_t(\underline{V}^k_{t+1}))$ at x where Benders' cut represents a single tangent plane (see [§3.2 VL12]). APM methods were extended to general distribution in [RPM21]. In [VL12], the authors provided a necessary and sufficient condition for a partition to be adapted (without Assumption (FSN)) as well as a geometric method to obtain a valid and adapted partition. In particular, the APM SDDP algorithm of [SS22] is a TFDP algorithm falling in the framework of Algorithm 4.

3.1.4 Risk-averse setting

We now briefly discuss extensions involving a maximization problem in the dynamic programming equation, arising for example from multistage risk-averse, robust or distributionally robust problems. Algorithm 4 can be adapted to such problems, by changing the definitions of the Bellman operators.

Further, in the risk-neutral case, Algorithm 4 is not symmetrical in its treatment of lower and upper approximations. As noted in Remark 3.1, for a minimization problem, in Algorithm 4, the forward phase Line 6 should be done using the lower approximations $\underline{V}^k_t$. More generally, one should use an *outer approximation* (that is under approximation for min sub-problems and upper approximations for max sub-problems) during the forward phase to be able to explore the state space. Thus, for those min-max problems, the computation of upper-approximations $\overline{V}^k_t$ is not optional.

Minimax problems. Baucke, Downward and Zakeri, in [BDZ17], presented a convergent problem-child algorithm to solve stochastic minimax dynamic programs. Although our framework of Algorithm 4 do not handle such minimax problem, we can extend it to do so. More precisely, we consider a problem where the decision maker chooses $x_t \in \mathcal{X}_t(x_{t-1}, y_{t-1}, \xi_t)$, and then an adversary chooses $y_t \in \mathcal{Y}_{t-1}(x_{t-1}, y_{t-1}, x_t, \xi_t)$. Thus, the Bellman operators are now defined as

$$\mathcal{B}_{t-1}(\widetilde{V})(x_{t-1}, y_{t-1}) = \mathbb{E}\big[\min_{x_t\in\mathcal{X}_t(x_{t-1},y_{t-1},\boldsymbol{\xi}_t)} \max_{y_t\in\mathcal{Y}_{t-1}(x_{t-1},y_{t-1},x_t,\boldsymbol{\xi}_t)} c_t(x_{t-1}, x_t, y_t, \boldsymbol{\xi}_t) + \widetilde{V}(x_t, y_t)\big]. \tag{3.2}$$

We can adapt the definition of reachable sets X^r_t and Y^r_t (see [§5 VL16] for details), and forward Bellman operators. Then, assuming that the reachable sets X^r_t and Y^r_t have finite dimensions d_x and d_y and diameter D, and that the objective functions are L-Lipschitz, the convergence and complexity results still hold developing on the ideas of [ZS22]. The upper bound on the number of effective iterations then becomes $K_\varepsilon \coloneqq \left(\frac{2DL}{\varepsilon-\gamma_\Sigma}\right)^{d_x+d_y}(T-1)^{d_x+d_y+1}$.

Robust Closely related, in [GTW19], Georghiou, Tsoukalas and Wiesemann presented the Robust Dual Dynamic Programming algorithm (RDDP) to solve multistage robust optimization problems. In such problems, instead of minimizing the expectation, we minimize considering the worst-case scenario $\xi_t \in \Xi_t$. Note that this robust setting can be seen as a particular case of minimax problems where we have deterministic random variables. The upper bound on the number of effective iterations then becomes $K_\varepsilon \coloneqq \left(\frac{2DL}{\varepsilon-\gamma_\Sigma}\right)^{d_x+d_\xi}(T-1)^{d_x+d_\xi+1}$.

risk-averse Multistage stochastic problems in the risk-averse setting are MSP where the expectation is replaced by a multiperiod risk measure.

We consider the risk-averse MSP Eq. (1.18), that is, a risk-averse MSP with nested coherent risk measures. Using the dual representation of risk measures (see Theorem 1.3), we get

$$\mathcal{B}_{t-1}(\widetilde{V}) = \max_{\boldsymbol{y}\in\mathcal{Q}A_\rho} \mathbb{E}_{\mathbb{P}}\Big[\min_{x_t\in\mathcal{X}_t(x_{t-1},\boldsymbol{\xi}_t)} \boldsymbol{y}\, c_t(x_{t-1},x_t,\boldsymbol{\xi}_t) + \boldsymbol{y}\widetilde{V}(x_t)\Big]. \tag{3.3}$$

Up to a slight change of notation, we can write this problem as a minimax problem. In particular, a sufficient condition to obtain convergence and complexity bounds for risk-averse MSP is that the set $\mathcal{Q}_\rho$ has a finite dimension and a finite diameter. For example, if Ω is finite, $\mathcal{Q}_\rho$ is contained in the space of random variables in Ω, isomorphic to a simplex of dimension $|\Omega|-1$ which has a finite diameter. More generally, if $\mathcal{Q}_\rho$ is contained in the convex hull of n random variables $(\boldsymbol{y}_k)_{k\in[n]}$, then $\mathcal{Q}_\rho$ has a finite diameter smaller than $\max_{k,\ell\in[n]}(\|\boldsymbol{y}_k-\boldsymbol{y}_\ell\|_\infty)$ and a finite dimension smaller than $n-1$, we can obtain complexity results similar to the risk-neutral case.

We now comment on the particular case of the average value at risk (see Definition 1.4). We cannot use the dual representation to derive complexity bounds as $\mathcal{Q}_{AV@R}$ has, in general, a non-finite dimension. However, note that in the min-formulation of AVAR, see Eq. (1.5b), since $AV@R_\alpha(\boldsymbol{z}) \leqslant \frac{\mathbb{E}_{\mathbb{P}}[\boldsymbol{z}]}{1-\alpha}$ the infimum on s over $\mathbb{R}$ can be replaced by a minimum on the compact interval $[0, \frac{1}{1-\alpha}\mathbb{E}_{\mathbb{P}}[\boldsymbol{z}]]$. To obtain an upper bound that does not depend on k and x_{t-1}, we set $\boldsymbol{z} = \min_{x_t\in\mathcal{X}(x_{t-1},\boldsymbol{\xi}_t)} c_t(x_{t-1},x_t,\boldsymbol{\xi}_t) + \underline{V}_t^k(x_t)$ then $\mathbb{E}_{\mathbb{P}}[\boldsymbol{z}]$ is upper bounded by $\min_{x_t\in X_t^\tau}\mathbb{E}\big[c_t(x_t,\boldsymbol{\xi}_t)+\overline{V}_t^1(x_t)\big]$ which has a finite value by assumption 3.2. Thus, MSP with nested average value at risk measure can also be handled and yield similar complexity results.

3.2 Convergence of TFDP algorithms

There are two main approaches to convergence theory of TFDP algorithms. The first, initiated by Philpott and Guan [PG08], and generalized in [ACdC20], argues that there is a finite number of cuts that can be generated and that they are generated at most once. Then, leveraging the fact that each scenario is sampled an infinite number of times, they prove the almost-sure convergence in a finite number of iterations, without any bound on this number.

Another path, pioneered in [VL3], (then reformulated and adapted to the risk-averse setting in [Gui16], and extended to abstract Bellman operators in [VL10]) is based on the compactness of the (reachable) state space and the Lipschitz property of the value function. These convergence proofs rely on the compactness of the (reachable) state space to extract converging subsequenceq. Technical derivations relying on Lipschitz regularity of the value function prove asymptotic convergence.

To go further than asymptotic convergence, two recent papers ([Lan20, ZS22]) quantify how the gap between estimated and true value function at a given point can be back propagated and extended to a ball of small radius r around this point. By bounding the number of non-intersecting balls of radius r that can be contained in the reachable state space they deduce a convergence speed. This is the approach that is extended in [VL16], and presented here.

First, it can be noted that all previous convergence proofs relied on the finite noise Assumption (FSN). Indeed, the first proofs used some sort of Borel-Cantelli lemma showing that every possible scenario is almost surely selected an infinite number of times during the forward pass[1]; while the more recent proof of [Lan20, ZS22] shows that a given scenario is reducing the gap, thus the expected gap is reduced by at least the (non-zero, but very small) probability of selecting this scenario multiplied by the gap reduction. The proof presented here does not require the finitely supported noise Assumption (FSN) replacing it with a nested Hoeffding lemma.

In this section, we give convergence and complexity results for various instances of Algorithm 4. In § 3.2.1, we first define the notion of *effective iteration* and deduce an upper bound on the number

[1]While these proofs are sound they remain quite theoretical as, in most practical cases, the number of possible scenarios is tens of orders of magnitude higher than the number of selected scenario, thus a given scenario has a very low probability to be selected once, let alone multiple times.

of effective iterations required by Algorithm 4 to get an ε-solution. We then distinguish between deterministic and randomized selection processes for the choice in Line 5 of the algorithm. For deterministic selection processes, namely the problem-child and explorative node selections, we show in § 3.2.2 that all iterations are effective. Finally, when the node selection is randomized, we show in § 3.2.3 the existence of a positive probability for an iteration to be effective. We then deduce a complexity bound on the expected number of iterations.

3.2.1 Bounding the number of effective iterations

We first recall that the value of Problem (MSP) can be written in a more concise form, (keeping in mind that ξ_1 is deterministic):

$$\text{val}\,(\text{MSP}) = \min_{x_1 \in \mathcal{X}_1(x_0,\xi_1)} c_1(x_0, x_1, \xi_1) + V_1(x_1). \tag{3.4}$$

Our aim is to show that, for some iteration k, the solution x_1^k is a ε-solution of Eq. (3.4), and the lower bound $\underline{V}_0(x_0)$ is ε-tight. Unfortunately, assumptions 3.1 to 3.4 and Assumption (SWI) are not enough to ensure convergence of Algorithm 4: we need a further assumption on the node selection process.

Regardless of node selection, we define the notion of *effective iteration*. Recall that γ_t^F, $\underline{\gamma}_t$, $\overline{\gamma}_t$ are errors in forward Bellman operator and approximation update (see Algorithm 4) at time $t \in [T]$, and $\underline{L}_t$ (resp. $\overline{L}_t$) are Lipschitz bounds on the cuts (resp. upper-approximation) at time t. In the remains of the section we consider a sequence $(\overline{V}_t^k, \underline{V}_t^k, x_k^t)_{t\in[T],k\in\mathbb{N}}$ produced by Algorithm 4.

We consider a sequence of positive real number $(\delta_t)_{t\in[T]}$, seen as precision diameter in the state space. We then construct an adequate sequence of value errors $(\varepsilon_t)_{t\in[T]}$[1].

An effective iteration, as defined in the following definition, is an iteration that produces at least a new trial point with a small gap, and is significantly different from past trial points (with a small gap).

Definition 3.3 (effective iteration)**.** *For $t \in [T-1]$ and $k \in \mathbb{N}$, we say that*

- x_t^k *is* ε_t-saturated*, if* $\overline{V}_t^k(x_t^k) - \underline{V}_t^k(x_t^k) \leqslant \varepsilon_t$*,*
- x_t^k *is* δ_t-distinguishable *if* $\|x_t^k - x_t^\kappa\| > \delta_t$ *for all* $\kappa < k$ *such that* x_t^κ *is* ε_t*-saturated,*
- *an iteration $k \in \mathbb{N}$ is* effective *if it generates either a new ε_t-saturated and δ_t-distinguishable point for at least one $t \in [T]$, or a ε_0-solution to the first-stage problem.*

As a first step tower showing a convergence speed, the following theorem bound the number of effective iterations required to obtain an ε_1 lower bound.

Theorem 6 (bound on effective iterations number)**.** *Let assumptions 3.1 to 3.4 and Assumption (SWI) be satisfied and $t \in [T-1]$, assume that $\delta_t \leqslant D_t$, and let*

$$\overline{K} \coloneqq \sum_{t=1}^{T-1} \left(\frac{D_t}{\delta_t} + 1 \right)^{n_t}. \tag{3.5}$$

After at most $\overline{K} + 1$ effective *iterations we have an ε_1-lower bound of Problem (MSP).*

$$\underline{V}_0^k(x_0) = c_1(x_1^k, \xi_1) + \underline{V}_1^k(x_1^k) \geqslant \text{val}\,(\text{MSP}) - \varepsilon_1. \tag{3.6}$$

[1] More precisely, $\varepsilon_{T-1} \coloneqq \underline{\gamma}_{T-1} + \overline{\gamma}_{T-1}$, $\varepsilon_t \coloneqq \varepsilon_{t+1} + (\overline{L}_{t+1} + \underline{L}_{t+1})(\delta_{t+1} + \eta_{t+1}) + \gamma_{t+1}^F + \underline{\gamma}_t + \overline{\gamma}_t$ for $t \in [T-2]$, and $\varepsilon_0 \coloneqq \varepsilon_1 + (\overline{L}_1 + \underline{L}_1)(\delta_1 + \eta_1) + \gamma_1^F$.

Remark 3.4. *Finally, although the theorems of this section state that we find an ε_0-optimal solution at stage 1, we have no guarantee that the approximations $\underline{V}_t^k$ converge to V_t. We cannot hope that these approximations converge to the true expected cost-to-go functions far from the optimal and reachable trajectories.*

Nevertheless, by considering the sets of points that are δ_t-close to every optimal and reachable trajectories, we could hope to have a convergence of strategies generated by $\mathcal{F}(\underline{V}_t^k)$ on those sets. If we add a finite diameter of the support of $\boldsymbol{\xi}_t$ and Lipschitz assumptions for $\boldsymbol{\xi}_t$, we are confident that the proof can be adapted. However, the general case looks harder and might require different ideas for proving complexity results for the convergence of strategies at every stage.

We now show that for some deterministic node selection methods, each iteration is effective.

3.2.2 Deterministic node selection

In this section, we present sufficient conditions for an iteration to be effective. Consequently, for two algorithms with deterministic node selections (namely problem-child node selection [BDZ17] and explorative node selection [Lan20]), we show that each iteration is effective, yielding a complexity result.

The following technical lemma, whose proof can be found in [Appendix C VL16], shows that if the new state x_t^k (resulting from the choice of ξ_t^k) is either i) far enough from the set of saturated points, or ii) yielding a large enough gap, then iteration k is effective.

Lemma 3.5. *We denote by d_t^k the distance function to the set of ε_t-saturated points until iteration k,* i.e., $d_t^k(x) \coloneqq \min_{\kappa<k|x_t^\kappa \text{ is } \varepsilon_t\text{-saturated}} \|x - x_t^\kappa\|$. *In particular, x_t^k is δ_t-distinguishable if and only if $d_t^k(x_t^k) > \delta_t$. Further, we denote $\boldsymbol{y}_t^k$ the possible next state starting from the current trajectory point and applying the current strategy,* i.e., $\boldsymbol{y}_t^k \coloneqq \mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \boldsymbol{\xi}_t)$.

Assume that assumptions 3.1 to 3.4 and Assumption (SWI) are satisfied. Let $k \in \mathbb{N}^*$. If, for all $t \in [T-1]$, at least one of the following inequalities is satisfied

$$\mathbb{E}\big[\overline{V}_t^{k-1}(\boldsymbol{y}_t^k) - \underline{V}_t^{k-1}(\boldsymbol{y}_t^k)\big] \leqslant \overline{V}_t^{k-1}(x_t^k) - \underline{V}_t^{k-1}(x_t^k) + (\overline{L}_t + \underline{L}_t)\eta_t, \tag{3.7a}$$

$$\mathbb{E}\big[d_t^k(\boldsymbol{y}_t^k)\big] \leqslant d_t^k(x_t^k) + \eta_t, \tag{3.7b}$$

then, iteration k is effective.

The following lemma shows that two deterministic node selection, i) the problem-child method of Baucke, Downward and Zakeri [BDZ17, BDZ18], which select the node yielding a state maximizing the current gap estimate; and ii) the explorative method of [Lan20], which select the node leading to the most distinguishable point, both satisfy the condition of Lemma 3.5. Thus, both node selection processes ensure that each iteration is effective.

Lemma 3.6. *We say that we have a* problem-child *node selection if for all $k \in \mathbb{N}^\star$, and $t \in [T-1]$, ξ_t^k is chosen such that it maximizes the current gap,* i.e.,

$$\xi_t^k \in \operatorname*{arg\,max}_{\xi \in \operatorname{supp}(\boldsymbol{\xi}_t)} \overline{V}_t^{k-1}\big(\mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \xi)\big) - \underline{V}_t^{k-1}\big(\mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \xi)\big). \tag{3.8a}$$

We say that we have an explorative *node selection if for all $k \in \mathbb{N}^\star$, and $t \in [T-1]$, ξ_t^k is chosen such that x_t^k maximizes the distance to previous ε_t-saturated points,* i.e.,

$$\xi_t^k \in \operatorname*{arg\,max}_{\xi \in \operatorname{supp}(\boldsymbol{\xi}_t)} d_t^k\big(\mathcal{F}_{t-1}(\underline{V}_{t+1}^{k-1})(x_{t-1}^k, \xi)\big). \tag{3.8b}$$

Then, with a problem-child *or an* explorative *node selection method, each iteration of Algorithm 4 is effective.*

As each iteration is effective, Theorem 6 provides a complexity bound given in the following corollary.

Corollary 3.7. *For simplicity, let the total error be* $\gamma_\Sigma \coloneqq \sum_{t=1}^{T-1} \underline{\gamma}_t + \overline{\gamma}_t + \gamma_t^F$ *and choose* n, D, L *such that, for all* $t \in [T-1]$, $n_t \leqslant n$, $D_t = D$, $\overline{L}_t = \underline{L}_t = L$. *Then, for every* $\varepsilon > \gamma_\Sigma$, *sufficiently small (e.g. such that* $\varepsilon \leqslant 2DL + \gamma_\Sigma$*), Algorithm 4 finds an* ε*-first-stage solution* x_1^k *within at most* $\overline{K}_\varepsilon$ *iterations where*

$$\overline{K}_\varepsilon \coloneqq \left(\frac{2DL}{\varepsilon - \gamma_\Sigma}\right)^n (T-1)^{n+1}. \tag{3.9}$$

3.2.3 Randomized algorithms

When the choice of ξ_t^k is made randomly, there is no guarantee that the iteration will be effective. Thus, we work in two steps: first, we show that there is a probability $p > 0$ for an iteration to be effective, then, by comparing the time to obtain an effective iteration to a geometric random variable of probability of success p, we deduce a bound on the expected number of iteration required to get an ε-optimal solution.

Remark 3.8 (Notational difficulty of randomized algorithm on stochastic problem)**.** *We are now considering a stochastic algorithm for solving the Problem* (MSP)*. Thus, there are two sources of randomness: the intrinsic* $\boldsymbol{\xi}_{[T]}$ *and the node selection* $\xi_t^k = \widetilde{\xi}_t^k$*. To distinguish both, we denote in bold random variables that are* $\boldsymbol{\xi}_{[T]}$ *measurable, with a tilde random variables that are* $(\widetilde{\xi}_t^k)_{t\in[T],k\in\mathbb{N}^*}$ *measurable (and with both if they are neither).*

For example, the trajectory determined during the forward phase $(\widetilde{x}_t^k)_{t\in[T]}$ *only depends on the past node selections, whereas the tentative points* $\widetilde{\boldsymbol{y}}_t^k$ *depends both on the past node selections and the actual realization of* $\boldsymbol{\xi}_t$*.*

Under Assumption (FSN), this discussion is usually avoided by representing the dependence on $\boldsymbol{\xi}_{[T]}$ *with a (finite) scenario tree, and indexing the variables by the tree nodes.*

Let $(\mathcal{A}^k)_{k\in\mathbb{N}^*}$ be the filtration such that $\mathcal{A}^k \coloneqq \sigma\big(\widetilde{\xi}_t^\kappa\big)_{t\in[T-1],\kappa\in[k]}$, and $\mathcal{A}^\infty = \bigcup_{k\in\mathbb{N}} \mathcal{A}^k$. In particular, a random variable measurable with $\mathcal{A}^k$ knows all node selection up to iteration k, which include, for example, $\underline{V}_t^k$ for all $t \in [T]$. Using a technical nested Hoeffding lemma, we give a lower bound over the probability of an iteration to be effective.

Lemma 3.9. *Assume that in Algorithm 4, Line 5, we draw* ξ_t^k *randomly according to the distribution of* $\boldsymbol{\xi}_t$*, and independently of all other* $\widetilde{\xi}_\tau^\kappa$ *as well as* $(\boldsymbol{\xi}_\tau)_{\tau\in[T-1]}$*.*

Then, for all iteration $k \in \mathbb{N}$ *of Algorithm 4 and all event* $A^{k-1} \in \mathcal{F}^{k-1}$ *such that* $\mathbb{P}\big[A^{k-1}\big] > 0$*, we have*

$$\mathbb{P}\Big[\textit{Iteration } k \textit{ is effective.} \,\Big|\, A^{k-1}\Big] \geqslant \prod_{t=1}^{T} \left(1 - e^{\frac{-2\eta_t^2}{D_t^2}}\right). \tag{3.10}$$

This lemma shows that the time to obtain an effective iteration is stochastically dominated by a binomial distribution, which yields the following bound on the expected number of iterations required for convergence.

Theorem 7. *Let assumptions 3.1 to 3.4 and Assumption (SWI) be satisfied and assume that in Line 5, we draw* ξ_t^k *randomly according to the distribution of* $\boldsymbol{\xi}_t$*, and independently of the previous* ξ_τ^κ*.*

Further, for simplicity, let the total error be $\gamma_\Sigma \coloneqq \sum_{t=1}^{T-1} \underline{\gamma}_t + \overline{\gamma}_t + \gamma_t^F$ *and choose* n, D, L *such that, for all* $t \in [T-1]$, $n_t \leqslant n$, $D_t = D$, $\overline{L}_t = \underline{L}_t = L$*.*

Then, for $\varepsilon > \gamma_\Sigma$ *, sufficiently small* (e.g., *such that* $\varepsilon \leqslant 4DL + \gamma_\Sigma$*), the expected number of iterations of Algorithm 4 required to find an* ε*-solution* x_1^k *to the first-stage problem is bounded by* $(T-1)\left(\frac{4DL(T-1)}{\varepsilon-\gamma_\Sigma}\right)^{n+2(T-1)}$.

Chapter 4

Exact upper bounds in multistage stochastic problem

Computing upper bounds of MSPs has always been challenging, which is somewhat surprising when coming from the deterministic world. Indeed, the easiest way to compute an upper bound is to evaluate the cost of an admissible solution, which is often easy in deterministic optimization. In risk-neutral stochastic optimization evaluating the cost of a solution requires computing an expectation, which is usually intractable for MSPs with a large horizon. For example for the hydroelectric valley problem of Example 1.17 it would require computing a sum over 10^{52} scenarios. The usual answer is to rely on the Monte Carlo method, which yields a statistical error. Unfortunately, this approach usually fails for risk-averse problems.

Further, in Chapter 3, we described a framework for Trajectory Following Dynamic Programming algorithms for solving (MSP) under the stagewise independence assumption. In this framework, even if it is mainly concerned with lower bounds, we also used upper bounds approximations of the value functions V_t. These upper bounds are sometimes simply theoretical, to enable the derivation of complexity results, and sometimes used as part of the algorithm itself, *e.g.,* for the problem-child node selection approach ([BDZ17]).

In this chapter, we are discussing how to compute and leverage such upper bounds. While some considerations hold true without the finite support Assumption (FSN), we assume that it holds true throughout the chapter. First, Section 4.1 briefly recalls the classical computation of upper bounds, either through statistical methods or by backward propagation. Then, Section 4.2 shows how we can use duality theory to write a Bellman recursion for the dual problem and then compute upper bounds. These results derive from [VL10] and [VL14]. Finally, Section 4.3 present possible uses for these upper bounds.

Note that there exist other methods, not discussed here, to determine upper bounds for MSPs. For example, partitioning methods coupled with Jensen's and Edmundson-Madansky's bounds yield efficient bounds, see [Kuh06] for a broad review and presentation. Moreover, we can approximate the information structure as in [MAB14, MP16] or do both simultaneously *e.g.,* in [Kuh08].

4.1 Primal computation of upper bounds

First, let's recall the setting considered here. We consider a, potentially risk-averse, (MSP) problem as in (1.21) with stagewise independent noise, and its associated Bellman operators

$$\hat{\mathcal{B}}_t(\widetilde{V}) = \begin{cases} \mathbb{R}^{n_{t-1}} \times \xi_t & \to \mathbb{R} \cup \{+\infty\} \\ (x_{t-1}, \xi_t) & \mapsto \inf\limits_{x_t \in \mathcal{X}(x_{t-1}, \xi_t)} c_t(x_{t-1}, x_t, \xi_t) + \widetilde{V}(x_t) \end{cases}, \tag{4.1a}$$

and

$$\mathcal{B}_t(\widetilde{V}) : x_{t-1} \mapsto \rho_t\big[\hat{\mathcal{B}}_t(\widetilde{V})(x_{t-1}, \boldsymbol{\xi}_t)\big]. \tag{4.1b}$$

where ρ_t are (conditional) coherent risk measures. The (MSP) is risk-neutral if all ρ_t are (conditional) expectations.

Recall that the associated Bellman equation reads

$$V_t = \mathcal{B}_t(V_{t+1}), \qquad \forall t \in [T], \qquad V_{T+1} \text{ given.} \tag{4.2}$$

and the value of the (MSP) is given by $V_1(x_0)$.

Our objective here is to compute upper bounds, and especially exact upper bounds, over the value of (MSP) $V_1(x_0)$, and more generally over the value function V_t.

4.1.1 Statistical upper bounds

A classical method to compute an upper bound for any minimization problem consists in evaluating the cost associated with an admissible solution. However, with multistage problem computing the cost associated with a policy is numerically challenging. The most classical approach then is to rely on statistical methods, especially straightforward in a risk-neutral setting.

risk-neutral problem

Consider a risk-neutral (MSP) problem, and an admissible policy, that is a measurable function that, given any current state x_{t-1} and noise realization ξ_t, returns an admissible state $x_t \in \mathcal{X}_t(x_{t-1}, \xi_t)$. A policy, for example, is induced by a collection of forward Bellman operators $\hat{\mathcal{F}}_t$, which is a way of solving each stage-problem (see § 1.3.2), and a collection of value functions approximation $(\widetilde{V}_t)_{t\in[T]}$. We can simulate such a strategy along any scenario. For strategy induced by approximate value functions, this corresponds to a forward pass in TFDP Algorithm 4, with random node selection.

For risk-neutral problems, the value of an admissible strategy is simply the expectation of the associated cost. While this expectation often cannot be computed in a reasonable time, it can almost[1] always be estimated by Monte Carlo see Algorithm 5.

Data: Random variables $\boldsymbol{\xi}_t$ simulator, initial state x_0, forward operators $\mathcal{A}_t$, Monte Carlo simulation number N, confidence level α.

for $k \in [N]$ **do**
  Draw a scenario $\xi^k_{t\ t\in[T]}$;
  Set $x^k_0 = x_0$;
  $C^k = 0$;
  for $t = 1 : T-1$ **do**
    Let $x^k_t = \mathcal{F}_{t-1}(\underline{V}^{k-1}_t)(x^k_{t-1}, \xi^k_t)$;
    $C^k \leftarrow C^k + c_t(x^k_{t-1}, x_t, \xi^k_t)$;
$C^{est} = \frac{\sum_{k=1}^N C^k}{N}$;
$\sigma^{est} = \sqrt{\frac{\sum_{k=1}^N (C^k - C^{est})^2}{N-1}}$;
$ub = C^{est} + z_{1-\alpha}\frac{\sigma^{est}}{\sqrt{N}}$

Algorithm 5: Computation of statistical upper bound ub by Monte Carlo with asymptotic confidence level of α, and $z_{1-\alpha}$ such that $\mathbb{P}(\boldsymbol{G} \leqslant z_{1-\alpha}) = 1-\alpha$ for $\boldsymbol{G} \sim \mathcal{N}(0,1)$.

This method is quite straightforward, and provides numerically good results. It can also be improved through a variety of methods (*e.g.,* variance reduction methods, quasi-Monte Carlo sampling methods...) Unfortunately the upper bound obtained remain statistical and given only with a confidence level. Further, risk-averse settings are more delicate.

[1] As the strategy is assumed to be admissible the associated cost is almost surely finite. However, for the Monte Carlo method to be relevant we should ensure that the cost has a finite variance, which is the case in most MSP settings.

risk-averse problem

The computation of statistical upper bounds for risk-averse problems is significantly harder than in the risk-neutral setting. Indeed, the statistical convergence theory exists but often requires, for nested risk measures, estimating the conditional risk measures at each time-step, meaning that the number of scenarios required is exponential in the number of stages see for example [Sha11]. In [KM15], Kozmík and Morton introduce an importance sampling scheme, dedicated to nested AVAR model, and leveraging a user-defined approximation function, that can improve convergence. Finally, [GSC21] suggest a statistical method for risk measure that can be written as the minimum of expected cost, with additional variables, that is, such that $\rho(\boldsymbol{X}) = \min_z \mathbb{E}_{\mathbb{P}}\big[\Psi(\boldsymbol{X}, z)\big]$ for some adequate function Ψ, AV@R (or convex combination of AV@R and expectation) being the most common example. At its heart, this methodology use a decision rule approach for the additional variable z, and then classical Monte Carlo estimation.

4.1.2 Backward propagation of convex upper bounds

Another approach consists in propagating upper bounds through convexity. The approach, which seems to have been proposed by [PdMF13], consists in using basic property of Bellman operators: monotonicity and conservation of convexity (see Proposition 1.24). Indeed, if, for some $t \in [T]$, $\overline{V}_{t+1}$ is convex and above V_{t+1} on the reachable states, *i.e.,* , $V_{t+1} \leqslant \overline{V}_{t+1} + \mathbb{I}_{X^r_{t+1}}$, then, for a convex Bellman operator, $\mathcal{B}_t(\overline{V}_{t+1})$ is also convex and above V_t on X^r_t.

More precisely, let $t \in [T]$, and choose N trial points $x^n_{t-1} \in X^r_{t-1}$. From a given upper bound function $\overline{V}_{t+1} \geqslant V_{t+1}$, compute $v^n_t = \mathcal{B}_t(\overline{V}_{t+1})(x^n_t)$. By monotonicity, we have $(x^n_t, v^n_t) \in \mathrm{epi}(V_t)$. By convexity of V_t we can define a function $\overline{V}_t \geqslant V_t$ such that its epigraph is $\mathrm{conv}(x^n_t, v^n_t) + \{0\}^{n_x} \times \mathbb{R}_+$, which can be expressed as

$$\overline{V}_t : x \mapsto \min_{\alpha \in \Delta_{n-1}} \Big\{ \sum_{n=1}^N \alpha_n v^n_t \quad | \quad \sum_{n=1}^N \alpha_n x^n_t = x \Big\}, \tag{4.3a}$$

where Δ_{n-1} is the simplex of $\mathbb{R}^n$, *i.e.,* , $\Delta_{n-1} = \big\{\alpha \in [0,1]^n \mid \sum_{n=1}^N \alpha_n = 1\big\}$. Equivalently, we have the dual expression

$$\overline{V}_t : x \mapsto \max_{\lambda, \mu} \quad \Big\{ \lambda^\top x + \mu \mid \lambda^\top x^n_t + \mu \leqslant v^n_t, \quad \forall n \in [N] \Big\}. \tag{4.3b}$$

The approach proposed in [PdMF13] consists in first running SDDP to obtain lower bounds, and uses the computed forward trajectory as trial points x^n_t. The upper bounds can thus be computed inductively Backward. Problem-child approaches (*e.g.,* [BDZ17]) use the same idea to iteratively refine the upper bounds.

We end with some refinement when we can ensure that V_t is L-Lipschitz continuous on the reachable set X^r_{t-1}.

Remark 4.1 (Lipschitz regularization)**.** *Let f and g be two proper functions of $\mathbb{R}^n$. Their inf-convolution is defined as $f \square g : x \mapsto \inf_{y \in \mathbb{R}^n} f(y) + g(x-y)$. A standard result of convex analysis ensures that $f^L \coloneqq f \square L\|\cdot\|_1$ is the largest L-Lipschitz function, for the L_1 norm, that is lower than f, also called the L-*Lipschitz regularization *of f.*

Assume that V_t is L-Lipschitz on X^r_{t-1}, then if $\overline{V}_t \geqslant V_t$ on X^r_{t-1}, then its L-Lipschitz regularization $\overline{V}^L_t \coloneqq \overline{V}_t \square L\|\cdot\|_1$ is also above V_t on X^r_{t-1}. If $\overline{V}_t$ is obtained by Eq. (4.3), then we can compute $\overline{V}^L_t(x)$ as follows:

$$\overline{V}^L_t(x) \quad = \min_{\alpha \in \Delta_{n-1}, y} \Big\{ \sum_{n=1}^N \alpha_n v^n_t + L\|x-y\|_1 \quad | \quad \sum_{n=1}^N \alpha_n x^n_t = y \Big\}, \tag{4.4a}$$

$$= \max_{\lambda, \mu} \quad \Big\{ \lambda^\top x + \mu \quad | \quad \|\lambda\|_\infty \leqslant L, \quad \lambda^\top x^n_t + \mu \leqslant v^n_t, \quad \forall n \in [N] \Big\}. \tag{4.4b}$$

4.2 Dual upper bounds

In this section, we choose to work with the linear setting (MSLP), as we heavily rely on convexity and duality. Some results can be extended to the non-linear convex setting but would require technical discussion to guarantee strong duality that is made simpler by the linear setting. For the same reason, we make the finite noise assumption Assumption (FSN).

Thus, we consider the following MSLP

$$\min_{\boldsymbol{x}_t, \boldsymbol{y}_t} \quad \rho_1\Big(\boldsymbol{c}_1^\top \boldsymbol{y}_1 + \rho_{2|\boldsymbol{\xi}_1}\Big(\cdots + \rho_{T|\boldsymbol{\xi}_{[T-1]}}(\boldsymbol{c}_T^\top \boldsymbol{y}_T)\Big)\Big) \tag{4.5a}$$

$$\text{s.t.} \quad \boldsymbol{A}_t\boldsymbol{x}_t + \boldsymbol{B}_t\boldsymbol{x}_{t-1} + \boldsymbol{T}_t\boldsymbol{y}_t = \boldsymbol{d}_t \qquad \forall t \in [T], \tag{4.5b}$$

$$0 \leqslant \boldsymbol{x}_t \leqslant \overline{x}_t,\ 0 \leqslant \boldsymbol{y}_t \leqslant \overline{y}_t \qquad \forall t \in [T], \tag{4.5c}$$

$$\boldsymbol{x}_t, \boldsymbol{y}_t \preccurlyeq \boldsymbol{\xi}_{[t]} \qquad \forall t \in [T]. \tag{4.5d}$$

where $\boldsymbol{\xi}_{[t]} = (\boldsymbol{\xi}_t)_{t\in[T]}$ is a sequence of stagewise independent, exogenous, finitely supported random variables. Further, $\boldsymbol{\rho}_{t|\xi_{[t]}}$ is a coherent risk measure conditional on the past noises $\boldsymbol{\xi}_{[t]}$, all equalities hold almost surely, and constraint (4.5d) is the non-anticipativity constraint.

In particular, contrary to Chapter 2, we chose to explicitly give a state $\boldsymbol{x}_t$ and a control variable $\boldsymbol{y}_t$. In a large part of the stochastic programming literature, the control y is considered part of the state x. However, as we have seen in Chapter 3, the complexity of TFDP algorithms depends on the dimension of the state $\boldsymbol{x}_t$ and not of the control $\boldsymbol{y}_t$.

In this section, we first tackle in § 4.2.1 the risk-neutral case by showing a Bellman recursion between the Fenchel transform of the expected-cost-to-go functions. For the risk-averse case, in § 4.2.2, we follow a different path, by writing Problem (4.5) in an extensive form, computing its dual and recognizing a Bellman recursion. We finally link with a transformation of the risk-adjusted cost-to-go functions.

4.2.1 risk-neutral case

In the risk-neutral case, that is when $\rho_{t|\boldsymbol{\xi}_{[t-1]}} \coloneqq \mathbb{E}[\cdot|\boldsymbol{\xi}_{[t-1]}]$, the cost-to-go function V_t satisfy the Bellman equation $V_t = \mathcal{B}_t(V_{t+1})$ with[1]

$$\mathcal{B}_t(R) : x_{t-1} \mapsto \min \quad \mathbb{E}\Big[\boldsymbol{c}_t^\top \boldsymbol{y}_t + R(\boldsymbol{x}_t)\Big] \tag{4.6a}$$

$$\text{s.t.} \quad \boldsymbol{A}_t\boldsymbol{x}_t + \boldsymbol{B}_t x_{t-1} + \boldsymbol{T}_t\boldsymbol{y}_t = \boldsymbol{d}_t, \tag{4.6b}$$

$$0 \leqslant x_{t-1} \leqslant \overline{x}_{t-1},\ 0 \leqslant \boldsymbol{x}_t \leqslant \overline{x}_t,\ 0 \leqslant \boldsymbol{y}_t \leqslant \overline{y}_t. \tag{4.6c}$$

The main idea of the approach presented here relies on being able to obtain a Bellman recursion for the Fenchel transform of the value function, that is an equation of the form $V_t^\star = \mathcal{B}_t^\ddagger(V_{t+1}^\star)$ where $\mathcal{B}_t^\ddagger$ is a Bellman operator. This is a straightforward, if tedious, computation. With this Bellman recursion, we can then run a TFDP algorithm, for example an SDDP algorithm, that computes outer approximations of $V_t^\star$ instead of V_t. Taking again the Fenchel transform, we obtain inner approximation of V_t. Unfortunately, this approach, developed in [VL10], is not completely straightforward. Indeed, the dual recursion obtained does not satisfy the compactness assumption required for SDDP (or more generally TFDP) to converge. We overcome this difficulty through Lipschitz regularization, which can also be interpreted as an exact penalization of some constraints. The remainder of this section brushes off the main steps toward this goal.

[1] For technical reasons we chose to incorporate a constraint on x_{t-1} in the definition of the Bellman operator, to easily bound its domain.

Remark 4.2 (Fenchel transform)**.** *Let E be a topological vector space and $E^\star$ it's topological dual. The* Fenchel transform *of a function $f : E \mapsto \overline{\mathbb{R}}$, is defined as*

$$f^\star(x^\star) := \sup_{x \in E} \left\langle x^\star, x \right\rangle - f(x). \tag{4.7}$$

We recall here a few elementary properties:

1. *$f^\star$ is a convex function;*
2. *if $f \leqslant g$ then $f^\star \geqslant g^\star$;*
3. *if f is L-Lipschitz, then $\|f^\star\|_\infty$ is bounded by L.*

Recall that (see Remark 1.21) a Bellman operator is an (abstract) *Linear Bellman operator* (LBO) if it has the form of (4.6), where constraint (4.6b) can be extended to any linear constraint on the random vector $(\boldsymbol{x}_t, \boldsymbol{y}_t)$. This allows to consider expectation constraints that naturally appear in the dual of a standard LBO (see below).

Recall that the reachable set X_t^r is the set of states that can be attained at time t (see Eq. (1.14)). The domain of $\mathcal{B}_t$, defined in (4.6), is the set of x_{t-1} such that there exists $(\boldsymbol{x}_t, \boldsymbol{y}_t)$ satisfying (4.6b)-(4.6c). We say that a LBO $\mathcal{B}_t$ is $\widetilde{V}$*-compatible*, for a proper polyhedral function $\widetilde{V}$, if for all $x_{t-1} \in \mathrm{dom}(\mathcal{B}_t)$, and all $\xi \in \mathrm{supp}(\boldsymbol{\xi}_t)$ the attainable states $x_t \in \mathcal{X}(x_{t-1}, \xi)$ are such that $\widetilde{V}(x_t) < +\infty$. We say that the sequence of LBO associated with (4.5) is *compatible* if $\mathcal{B}_t$ is compatible with V_{t+1} for all t, or, in other words, if, for all $t \in [T]$, $X_t^r \subset \mathrm{dom}(\mathcal{B}_t)$. We can show that, for any proper polyhedral function R such that $\mathcal{B}_t$ is R-compatible,

$$\Big[\mathcal{B}_t(R)\Big]^\star = \mathcal{B}_t^\ddagger\Big[R^\star\Big], \tag{4.8}$$

where

$$\begin{aligned}
\mathcal{B}_t^\ddagger(Q) : x_{t-1}^\star \mapsto \inf_{\boldsymbol{\lambda}, \boldsymbol{\zeta}, \boldsymbol{x}_t^\star} \quad & \mathbb{E}\left[\boldsymbol{d}^\top \boldsymbol{\lambda} + \overline{x}_t \boldsymbol{\zeta}^x + \overline{y}_t \boldsymbol{\zeta}^y + \overline{x}_{t-1}(\zeta^{x-})^+ + Q(\boldsymbol{x}_t^\star)\right] && \text{(4.9a)} \\
\text{s.t.} \quad & \boldsymbol{c} + \boldsymbol{T}_t^\top \boldsymbol{\lambda} + \boldsymbol{\zeta}^y \geqslant 0, && \text{(4.9b)} \\
& \boldsymbol{x}_t^\star + \boldsymbol{A}_t^\top \boldsymbol{\lambda} + \boldsymbol{\zeta}^x \geqslant 0, && \text{(4.9c)} \\
& x_{t-1}^\star = \mathbb{E}\big[\boldsymbol{B}_t^\top \boldsymbol{\lambda}\big] + \zeta^{x-}, && \text{(4.9d)} \\
& \boldsymbol{\zeta}^x, \boldsymbol{\zeta}^y \geqslant 0. && \text{(4.9e)}
\end{aligned}$$

From this, we have that $V_t^\star = \mathcal{B}_t^\ddagger\Big[V_{t+1}^\star\Big]$. Unfortunately, this recursion does not satisfy our need as $\mathcal{B}_t^\ddagger$ is not *compact*, more precisely we are lacking bounds constraints on the dual variables.

This problem is solved by compacting the dual Bellman operator. More precisely, through technical lemma (see [VL10], Proposition 2.8) we can guarantee that V_t is L_t-Lipschitz, on X_t^r for the L_1 norm, and hence that we can constrain the dual variable to be bounded by L_t. Thus, we define the compactified operator $\mathcal{B}_{t,L_t}^\ddagger(Q)$ which adds to (4.9) the following bound constraint

$$\|x_t^\star\|_\infty \leqslant L_t. \tag{4.9f}$$

Remark 4.3 (Lipschitz regularization and dual bound)**.** *Let f be a proper convex function. Recall that $f^L := f \square L\|\cdot\|_1$ is the largest L-Lipschitz function, for the L_1 norm, that is lower than f. Standard convex analysis ensure that $(f^L)^\star = f^\star + \mathbb{I}_{B_\infty(0,L)}$.*

Note that if V_t is L_t-Lipschitz on X_t^r, then $V_t^{L_t} = V_t$ on X_t^r, and taking the Fenchel transform yields a sequence of compatible, compact, abstract LBOs such that

$$\begin{cases} V_{T+1}^\star &= \mathbb{I}_0 \\ V_t^\star &= \mathcal{B}_{t,L_t}^\ddagger(V_{t+1}^\star) \end{cases} \tag{4.10}$$

With this Bellman recursion, any TFDP algorithm, for example SDDP, can be run over (4.10) to obtain a lower bound over $V_t^\star$, which in turn yields (after Fenchel transform), an upper bound of V_t.

We end with small comments on the required elements to run a TFDP algorithm on the dual recursion.

Remark 4.4 (Compacity of the dual operator)**.** *In the primal problem, relatively complete recourse yields compatibility (up to some additional explicit constraint) and compactness is ensured by* (4.6c)*. However, compactness of the domain of V_t directly implies Lipschitz regularity of $V_t^\star$, and in particular that the domain of $V_t^\star$ is not compact. To circumvent this difficulty we have to consider the Lipschitz regularization $V_t^{L_t}$, whose Fenchel transform has a compact domain. In other words, bounding the dual variables is equivalent to considering Lipschitz's regularization of the primal. This technicality is unavoidable to ensure the compactness of the dual.*

Remark 4.5 (Dual Relatively complete recourse)**.** *Compatibility of the dual Bellman operator, or relatively complete recourse in the dual, was not discussed here. In fact, this can be a non-trivial difficulty in some classical settings (see [GSC21]). Here, this assumption is satisfied by having explicitly written bounds on the state and control variables in* (4.6c)*. These explicit bounds result in dual variables ζ which can equivalently be interpreted as slack variables for the dual problem with exact penalization cost given by the primal upper bounds see Eq.* (4.9a)*.*

4.2.2 risk-averse case

As discussed in § 4.1.1, risk-neutral upper bounds can easily be estimated through statistical means, while risk-averse upper bounds are more challenging. We have thus extended the previous approach to encompass polyhedral risk-averse setting in [VL14]. We take a different approach than in § 4.2.1 which dodges some technical difficulties and offers another understanding of the method. Indeed, we consider an extensive formulation of Problem (4.5), seen as a large linear program. Standard linear programming duality theory yields an extended formulation of the dual problem, we then recognize a Bellman recursion to solve it. The main trick here is to consider a dual function with two arguments: the dual state value, and the worst-case probability. We can then apply a TFDP algorithm, *e.g.,* SDDP, on this dual, concave, formulation to obtain upper bounds of the dual problem, which are upper bounds of the primal problem through strong duality of linear programs.

We present the main arguments here.

Polyhedral risk measures and duality

We consider a *polyhedral risk measure* ρ, that is, a coherent risk measure of the form

$$\rho : \boldsymbol{t} \mapsto \sup_{\mathbb{Q}\in\mathcal{Q}} \mathbb{E}_{\mathbb{Q}}[\boldsymbol{t}] = \max_{k\in[K]} \{\mathbb{E}_{\mathbb{Q}^k}[\boldsymbol{t}]\}, \tag{4.11}$$

where $\mathcal{Q} = \mathrm{conv}(\{\mathbb{Q}^k\}_{k\in[K]})$. Polyhedral risk measures can be either chosen as interpretable risk-measures (e.g. $AV@R$ in a finite setting) or as the worst case among a set of probabilities estimated by various experts. Since we don't assume a reference probability, we resort to describing the extremal risk measures, which may be very numerous.

We denote the elements of the support of $\boldsymbol{\xi}$ by $\xi_1, \dots, \xi_J$, and $\mathbb{Q}^k[\boldsymbol{\xi} = \xi_j] = q_j^k$. Now, $V_t(x_{t-1})$ (as

recalled in Eq. (4.2)) is given by:

$$\begin{aligned}
\inf_{\boldsymbol{x},\boldsymbol{y};z,\boldsymbol{\theta}} \quad & z & & & (4.12)\\
\text{s.t.} \quad & z \geqslant \sum_{j\in[J]} q_j^k \theta_j & \forall k, \quad & [\phi_k] \\
& \theta_j \geqslant c_j^\top y_j + V_{t+1}(x_j) & \forall j, \quad & [\gamma_j] \\
& A_j x_j + B_j x_{t-1} + T_j y_j = d_j & \forall j, \quad & [\lambda_j] \\
& x_j, y_j \geqslant 0 & \forall j, \quad & [\mu_j, \nu_j] \\
& x_j \leqslant \overline{x}_t, \quad y_j \leqslant \overline{y}_t & \forall j. \quad & [\zeta_j^x, \zeta_j^y]
\end{aligned}$$

Using the dual multipliers denoted with Greek letters as indicated, we obtain the following dual problem

$$\begin{aligned}
\sup_{\substack{\phi_k,\gamma_j,\lambda_j,\\ \mu_j,\zeta_j^y,\zeta_j^x}} \sum_{j\in[J]} \Big[& \lambda_j^\top (B_j x_{t-1} - d_j) - \overline{x}_t \zeta_j^x - \overline{y}_t \zeta_j^y & (4.13)\\
& \qquad + \inf_{x_j} \gamma_j V_{t+1}(x_j) + (A_j^\top \lambda_j - \mu_j + \zeta_j^x)^\top x_j \Big] \\
\text{s.t.} \quad & \sum_k \phi_k = 1, \qquad \phi_k \geqslant 0, \\
& \sum_k \phi_k q_j^k = \gamma_j & \forall j, \\
& \gamma_j c_j + T_j^\top \lambda_j + \zeta_j^y \geqslant 0 & \forall j, \\
& \mu_j \geqslant 0 & \forall j.
\end{aligned}$$

The constraints on ϕ_k are equivalent to describing the vector of γ_j's as a convex combination of the extreme probabilities $\mathbb{Q}^k$. Therefore, one can rewrite problem (4.13) to include the constraint $\{\gamma_j\}_{j\in[J]} \in \mathcal{Q}$ instead of the first two lines. This, shows that the variables γ_j correspond to one supporting the probability of the risk measure ρ. In particular, if a given scenario is *effective*, in the sense of [RBHdM19], then there exists an optimal γ which charge this scenario.

Multistage risk-averse problem duality

We now extend the duality to the full multistage problem. In the stagewise independent setting, we let Ω_t be the set of all possible realizations of ξ_t, and the risk measure ρ_t is defined by $\rho_t = \sup_{\mathbb{Q}\in\mathcal{Q}_t} \mathbb{E}_{\mathbb{Q}}[\cdot]$, for a polyhedral subset $\mathcal{Q}_t$ of probability measures on Ω_t. The tree $\mathcal{T}$ describing the stochastic process is such that each node n of depth t is associated with a possible value of $\boldsymbol{\xi}_{[t]}$. For any node n, the set of its children is denoted by C_n, and $\mathcal{L}$ is the set of leaves of $\mathcal{T}$.

We introduce variables z_n to stand for the risk-adjusted value of our problem starting from node n, and θ_m represents the cost-to-go following the branch of node $m \in C_n$. To reduce the notational burden, we assume that, for all t, $\rho_t = \rho$. Then, the risk-averse problem (4.5), with value $V_{n_0}(\widetilde{x}_{n_0})$, can be written as the following linear program:

$$
\begin{array}{lll}
\min & z_0 & \\
\text{s.t.} & \displaystyle\sum_{m\in C_n} q_m^k \theta_m \leqslant z_n & \forall n \in \mathcal{T}\backslash\mathcal{L}, \forall k \in [K], \quad [\Phi_n^k] \\
& c_m^\top y_m + z_m \leqslant \theta_m & \forall m \in \mathcal{T}\backslash\{n_0\}, \quad [\gamma_m] \\
& A_m x_m + B_m \widetilde{x}_n + T_m y_m = d_m & \forall n \in \mathcal{T}\backslash\mathcal{L}, \forall m \in C_n, \quad [\lambda_m] \\
& z_\ell = 0 & \forall \ell \in \mathcal{L}, \quad [\eta_\ell] \\
& x_n = \widetilde{x}_n & \forall n \in \mathcal{T}\backslash\mathcal{L}, \quad [\pi_n] \\
& x_m \geqslant 0, y_m \geqslant 0 & \forall m \in \mathcal{T}\backslash\{n_0\}, \quad [\mu_m, \nu_m] \\
& \widetilde{x}_m \leqslant \overline{x}_m,\ y_m \leqslant \overline{y}_m & \forall m \in \mathcal{T}\backslash\{n_0\}, \quad [\zeta_m^x, \zeta_m^y]
\end{array}
\tag{4.14}
$$

where $\widetilde{x}_{n_0}$ is a parameter and not a variable, and we add the equalities $x_n = \widetilde{x}_n$ to highlight the time dynamics. Defining $\gamma_{n_0} = 1$, the linear programming dual of the above problem is

$$
\begin{array}{lll}
\displaystyle\sup_{\Phi,\gamma,\pi,\lambda} & \pi_{n_0}^\top \widetilde{x}_{n_0} - \displaystyle\sum_{m\in\mathcal{T}\backslash\{n_0\}} \lambda_m^\top d_m - \overline{x}_m^\top \zeta_m^x - \overline{y}_m^\top \zeta_m^y & \\
& \displaystyle\sum_{k\in[K]} \Phi_n^k = \gamma_n & \forall n \in \mathcal{T}\backslash\mathcal{L}, \quad [z_n] \\
& \displaystyle\sum_{k\in[K]} \Phi_n^k q_m^k = \gamma_m & \forall n \in \mathcal{T}\backslash\mathcal{L}, \forall m \in C_n, \quad [\theta_m] \\
& \pi_n = \zeta_n^x + \displaystyle\sum_{m\in C_n} B_m^\top \lambda_m & \forall n \in \mathcal{T}\backslash\mathcal{L}, \quad [\widetilde{x}_n] \\
& \pi_\ell = \zeta_\ell^x & \forall \ell \in \mathcal{L}, \quad [z_\ell] \\
& \pi_m + A_m^\top \lambda_m \geqslant 0 & \forall m \in \mathcal{T}\backslash\{n_0\}, \quad [x_m] \\
& \gamma_m c_m + T_m^\top \lambda_m + \zeta_m^y \geqslant 0 & \forall m \in \mathcal{T}\backslash\{n_0\}, \quad [y_m] \\
& \Phi_n^k \geqslant 0, \gamma_n \geqslant 0 & \forall n \in \mathcal{T}, \\
& \zeta_m^x \geqslant 0, \zeta_m^y \geqslant 0 & \forall m \in \mathcal{T}\backslash\{n_0\}.
\end{array}
\tag{4.15}
$$

Note that Φ_n can be seen as barycentric coordinates of the extreme points of $\mathcal{Q}$. Thus, the first two constraints can be more compactly written as $(\gamma_m)_{m\in C_n} \in \gamma_n \mathcal{Q}$.

By backward recursion, this problem can be solved through the following recursive equations, where, for all leaves $\ell \in \mathcal{L}$, and $n \in \mathcal{T}\backslash\mathcal{L}$,

$$
D_\ell(\pi_\ell, \gamma_\ell) = \quad \mathbb{I}_{\{\pi_\ell \geqslant 0\}} + \zeta_\ell^\top \overline{x}_\ell, \tag{4.16a}
$$

$$
\begin{aligned}
D_n(\pi_n, \gamma_n) = \quad & \sup_{\zeta_n^x, \pi_m, \gamma_m, \lambda_m, \zeta_m^y} \mathbb{1}_{\{n=n_0\}} \pi_{n_0}^\top \widetilde{x}_{n_0} - \overline{x}_n^\top \zeta_n^x + \\
& \qquad \sum_{m\in C_n} -\lambda_m^\top d_m - \overline{y}_m^\top \zeta_m^y + D_m(\pi_m, \gamma_m) \\
& \text{s.t.} \quad (\gamma_m)_{m\in C_n} \in \gamma_n \mathcal{Q}, \\
& \qquad \zeta_n^x + \sum_{m\in C_n} B_m^\top \lambda_m = \pi_n, \\
& \qquad \pi_m + A_m^\top \lambda_m \geqslant 0, \qquad \forall m \in C_n, \\
& \qquad \gamma_m c_m + T_m^\top \lambda_m + \zeta_m^y \geqslant 0, \qquad \forall m \in C_n, \\
& \qquad \zeta_m^x \geqslant 0, \zeta_m^y \geqslant 0, \qquad \forall m \in \mathcal{T}\backslash\{n_0\}.
\end{aligned}
\tag{4.16b}
$$

By the independence assumption, a backward induction shows that $D_n = D_{n'}$ for all nodes n and n' of the same depth. Thus, defining, we obtain the following recursion for the dual value functions:

$$D_T(\pi_T, \gamma_T) = \quad \mathbb{I}_{\{\pi_T \geqslant 0\}} + \zeta_T^\top \overline{x}_T, \tag{4.17a}$$

$$\begin{aligned} D_t(\pi_t, \gamma_t) = \sup_{(\gamma_j, \lambda_j, \pi_j, \zeta_j^y)_{j\in[J_t]}, \zeta^x} & -\overline{x}_t^\top \zeta^x + \sum_{j\in[J_t]} -d_j^\top \lambda_j - \overline{y}_{t+1}^\top \zeta_j^y - \overline{x}_t^\top \zeta^x \\ & + \sum_{j\in[J_t]} -d_j^\top \lambda_j - \overline{y}_{t+1}^\top \zeta_j^y + D_{t+1}(\pi_j, \gamma_j) \end{aligned} \tag{4.17b}$$

$$\begin{aligned} \text{s.t.} \quad & (\gamma_j)_{j\in[J_t]} \in \gamma_t \mathcal{Q}, \\ & \zeta^x + \sum_{j\in[J_t]} B_j^\top \lambda_j = \pi_t \\ & \pi_j + A_j^\top \lambda_j \geqslant 0, && \forall j \in [J_t], \\ & \gamma_j c_j + T_j^\top \lambda_j + \zeta_j^y \geqslant 0, && \forall j \in [J_t], \\ & \zeta_j^y \geqslant 0, \quad \zeta^x \geqslant 0. \end{aligned}$$

This decomposition satisfies the RCR conditions. Indeed, for every π_t and every $\gamma_t \geqslant 0$, any $\gamma \in \gamma_t \mathcal{Q}$ and $\lambda = 0$ are admissible, using slack variable ζ^x as needed. Then, π_j are chosen to satisfy the inequalities $\pi_j + A_j^\top \lambda_j \geqslant 0$, and the remaining constraints can be adjusted using ζ_j^y.

Remark 4.6. *Relatively complete recourse in a dual formulation is not guaranteed (see for example [GSC19]). In our setting, the explicit upper bounds over x and y ensure RCR. The existence of such upper bounds is equivalent to the existence of exact penalization coefficients in the dual, which is the tool used in [GSC19] to deal with this difficulty. Alternatively, we could incorporate feasibility cuts in the algorithm.*

Bounding the dual state

With our boundedness assumption, we have relatively complete recourse in the dual. To prove convergence, we still need to ensure that the dual state remains bounded through a compactification process akin to the risk-neutral case.

By assumption, we know that there exists an optimal primal solution. Further, by linear programming duality, we know that there exists an optimal dual solution. The marginal interpretation of the Lagrange multiplier π (see Problem (4.14)) states that, for each node, the optimal dual π_n is a subgradient of the primal value function *for* $\gamma_n = 1$. In particular, π_n/γ_n can be bounded by the Lipschitz constant of the primal value function V_n. In the independent setting, assuming that V_t is L_t-Lipschitz continuous on its domain, we can add the constraint $|\pi_j| \leqslant \gamma_j L_{t+1}$ to (4.17) for each j, without changing its value.

Therefore, we use the compactified recursion presented in (4.18). Since it has RCR and bounded states, the SDDP algorithm on this recursion converges.

$$\begin{aligned} D_t(\pi_t, \gamma_t) = \inf_{\zeta^x, \gamma_j, \lambda_j, \pi_j, \zeta_j^y} \quad & \overline{x}_t^\top \zeta^x + \sum_{j\in[J]} d_j^\top \lambda_j + \overline{y}_{t+1}^\top \zeta_j^y + D_{t+1}(\pi_j, \gamma_j) \\ \text{s.t.} \quad & \gamma \in \gamma_t \mathcal{Q}, \\ & \zeta^x + \textstyle\sum_j B_j^\top \lambda_j \geqslant \pi_t, \\ & \pi_j + A_j^\top \lambda_j \geqslant 0 && \forall j \in [J_t], \\ & \gamma_j c_j + T_j^\top \lambda_j + \zeta_j^y \geqslant 0, && \forall j \in [J_t], \\ & |\pi_j| \leqslant \gamma_j L_{t+1}, && \forall j \in [J_t], \\ & \zeta^x \geqslant 0,\ \zeta_j^y \geqslant 0, && \forall j \in [J_t]. \end{aligned} \tag{4.18}$$

Dual risk-averse Bellman operator

We start with some convex analysis recalls.

Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a proper lower semicontinuous convex function. Recall (see [Com18] for more details) that the *perspective function* of f, denoted $\widetilde{f}$, is a convex, lower-semicontinuous function of $\mathbb{R}^{1+n}$, such that

$$\widetilde{f} : \mathbb{R} \times \mathbb{R}^n \to (-\infty, +\infty] \tag{4.19}$$
$$(\gamma, x) \mapsto \begin{cases} \gamma f(x/\gamma), & \text{if } \gamma > 0 \\ \mathrm{rec}(f)(x), & \text{if } \gamma = 0 \\ +\infty & \text{otherwise} \end{cases}$$

where $\mathrm{rec}(f)$ is the *recession function* of f defined as

$$\mathrm{rec}(f) : \mathbb{R}^n \to]-\infty, +\infty] : x \mapsto \lim_{t \to +\infty} \frac{f(z + tx)}{t}\ , \tag{4.20}$$

where z can be chosen as any point in the domain of f. Both the recession and the perspective function of proper lower semicontinuous convex function are proper lower semicontinuous convex function as well.

Inspired by the recurrences in (4.13), we introduce the *coperspective function*. Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$. The coperspective of f is the perspective of the Fenchel conjugate, that is $(f^\star)^\sim$, that we denote $f^\boxtimes$. In particular, for $\psi \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}_{++}$, we have

$$f^\boxtimes(\psi, \gamma) \coloneqq \sup_{x \in \mathbb{R}^n} \psi^\top x - \gamma f(x). \tag{4.21}$$

Remark 4.7. *The coperspective is jointly convex in ψ, γ, lower semicontinuous, and a positively homogeneous function of degree 1: for all $t > 0$,*

$$f^\boxtimes(t \cdot \psi, t \cdot \gamma) = t \cdot f^\boxtimes(\psi, \gamma).$$

Further, cuts for a convex function and its perspective are essentially equivalent. If $f(x) \geqslant f(x_0) + g^\top(x - x_0) = \theta + g^\top x$, then

$$\begin{aligned} \widetilde{f}(x,t) = t \cdot f(x/t) &\geqslant t f(x_0) + t g^\top (x/t - x_0) \\ &\geqslant t f(x_0) + g^\top (x - t \cdot x_0) \\ &\geqslant \theta \cdot t + g^\top x \end{aligned}$$

Similarly, if $\widetilde{f}(x,t) \geqslant \theta \cdot t + g^\top x + \beta$, then $f(x) \geqslant g^\top x + \theta + \beta$. Note that if the cut for $\widetilde{f}$ is exact we can assume $\beta = 0$.

Consider a polyhedral risk measure ρ and the associated risk-averse Bellman operator $\mathcal{B}$ that, to any cost-to-go function V and initial state x_{t-1} associates the value of Problem (4.12).

The coperspective of $\mathcal{B}(V)$ can be calculated using (4.13). Leveraging positive homogeneity, for $\psi_0 \in \mathbb{R}^n$ and $\gamma_0 > 0$, we get that $\mathcal{B}(V)^\boxtimes(\psi_0, \gamma_0)$ is given by

$$\begin{aligned} \sup_{x_0}\ & \psi_0^\top x_0 \\ & + \inf_{\substack{\gamma, \lambda, \mu \\ \zeta^x, \zeta^y}} \sum_{j \in [J]} \lambda_j^\top (d_j - B_j x_0) + {\zeta_j^y}^\top \overline{y}_{t+1} \\ & \qquad + {\zeta_j^x}^\top \overline{x}_{t+1} + V^\boxtimes(\mu_j - A_j^\top \lambda_j - \zeta_j^x, \gamma_j) \\ & \text{s.t.} \quad \gamma \in \gamma_0 \mathcal{Q} \\ & \qquad \gamma_j c_j + \zeta_j^y + T_j^\top \lambda_j \geqslant 0 \qquad \forall j \\ & \qquad \mu_j, \zeta_j^x, \zeta_j^y \geqslant 0. \end{aligned} \tag{4.22}$$

Note that, if V is polyhedral, so are its Fenchel dual and its perspective. Thus, by linear programming duality, we can interchange $\sup$ and $\inf$ to obtain

$$
\begin{aligned}
[\mathcal{B}(V)]^{\boxtimes}(\psi_0, \gamma_0) = \inf_{\substack{\gamma,\lambda \\ \zeta^x,\zeta^y}} \quad & \sum_{j\in[J]} \lambda_j^\top d_j + \zeta_j^{y\top} \overline{y}_{t+1} + \zeta_j^{x\top} \overline{x}_{t+1} + V^{\boxtimes}(\psi_j, \gamma_j) \\
\text{s.t.} \quad & \sum_j B_j^\top \lambda_j = \psi_0 \\
& \gamma \in \gamma_0 \mathcal{Q} \\
& \gamma_j c_j + \zeta_j^y + T_j^\top \lambda_j \geqslant 0 \quad \forall j \\
& \psi_j + \zeta_j^x + A_j^\top \lambda_j \geqslant 0 \quad \forall j.
\end{aligned} \tag{4.23}
$$

This equation also defines a risk-*neutral* LBO $\mathcal{B}^{\boxtimes}$ that takes a *homogeneous* recourse function $V^{\boxtimes}$ and returns another homogeneous convex function of the same dimension. We call this operator the *projective dual Bellman operator* associated with $\mathcal{B}$.

Comparing (4.17) and (4.23), we notice the decomposition is not done at the same time-step for all variables: in the first one, ζ^x is a single variable, relaxing the incoming dual state constraint; whereas in the second, it relaxes the outgoing dual state constraint. Substituting $\pi_j = \gamma_j + \psi_j$, we obtain the following proposition, linking the coperspectives of the primal value functions with the value functions of the dual problem.

Proposition 4.8. *For $t \in [T]$, if the dual value function D_t is defined by* (4.17)*, and V_t is the primal value function*

$$
D_t(\pi_t, \gamma_t) = -\inf_{\zeta_t^x + \psi_t = \pi_t, \zeta_t^x \geqslant 0} \overline{x_t}^\top \zeta_t^x + V_t^{\boxtimes}(\psi_t, \gamma_t).
$$

In particular, D_t is a concave, positively homogeneous, one-sided Lipschitz regularization of $V_t^{\boxtimes}$.

Further, the value of primal Problem (4.5) *is* $\sup_{\pi_0} \quad \pi_0^\top x_0 + D_0(\pi_0, 1)$.

4.3 Using upper bounds

In this section, we briefly discuss a few uses of upper bounds for MSPs, in addition to simply giving a cost estimate: to stop the algorithm, to modify the algorithm and to construct new admissible policies.

4.3.1 Stopping test

For any algorithm, we need to have a *stopping rule*, which is a test that stops the algorithm. In some cases (*e.g.,* shortest path problem) the optimization algorithm stop because it reached the optimal solution of the problem at hand. Most commonly, either because the convergence is asymptotic or too slow, the algorithm is stopped before. A stopping test can be very pragmatic (*e.g.,* running time, number of iterations), linked with a slowing of the algorithm (*e.g.,* small gradient norm) or more mathematically precise, typically by guaranteeing some property of the returned solution. For example, providing upper and lower bounds allow us to precisely characterize the value of the problem, and the algorithm can be run until we get the desired precision ε. If in addition the upper bound also bound the cost associated with the current solution we can guarantee that the solution returned is ε-optimal.

When to stop an SDDP algorithm is often a difficult question. The original suggestion [PP91] was to stop when the exact lower bound reached the interval confidence of the upper bound obtained by Monte Carlo with the current solution. This suggestion is mathematically flawed as an imprecise Monte Carlo estimation results in an earlier stopping[1]. Shapiro [Sha11] improved this idea by suggesting stopping when the difference between the (exact) lower bound and the upper bound of the Monte Carlo confidence interval is small enough. This stopping test is mathematically sound[2], but can slow the algorithm.

[1] Of course, good practitioners used it with a reasonable number of scenarios, keeping it relevant.

[2] Although one should be careful as to what this stopping test guarantees. Repeating the test means that the confidence decrease.

Shapiro suggested using past forward iterations, instead of new simulations, as a cheap proxy for this upper bound estimation, reporting good numerical results.

In addition to the mathematical quality of the stopping test, we know from theory (see Chapter 3) that the convergence of TFDP algorithm is slow. In practice, we observe that we improve the quality of the lower bound up to a certain instance-dependent point after which the algorithm progress very slowly. Hence, [HdMDMF11] discussed other statistical stopping criteria that test that the lower bound is not improving anymore within any reasonable length of time.

In [VL10] we argue that upper bounds, especially exact upper bounds, can be used to design precise and reliable stopping tests. Indeed, we run both a primal and dual SDDP algorithm, and stop when the difference between the exact upper and lower bound are smaller than ε. The same obviously holds true for the risk-averse setting.

4.3.2 Algorithms variations

Computing upper bounds can be used to derive variations of the SDDP algorithm, which falls into the TFDP framework. Recall that the forward phase of a TFDP Algorithm 4 consists in a *node selection* (choosing $\xi_t^k \in \operatorname{supp}(\boldsymbol{\xi}_t)$), and a forward step using the current lower approximation for a given forward operator ($x_t^k = \mathcal{A}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \xi_t^k)$). Hence, upper bounds can be used in at least two ways: in the node selection process, and in the forward operator definition.

Problem-child node selection

When maintaining both a lower and upper bound, problem-child node selection (*e.g.,* [BDZ17]) consists in selecting the node leading to the worst possible gap. That is, given a current state x_{t-1}^k, we compute, for all $\xi \in \operatorname{supp} \boldsymbol{x}i_t$, the potential next state $x_t^k = \mathcal{A}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \xi)$. For each potential next state we compute the gap $\overline{V}_t(x_t^k) - \underline{V}_t(x_t^k)$. The state selected to go forward is the one with the maximal gap.

This node selection process has been shown to be efficient in some settings, while generally being slower than random node selection. However, it is more flexible than random node selection, as it can be adapted to some non-convex settings (e.g. [DDB20]).

Regularization

SDDP algorithm can be seen as a multistage extension of Kelley's cutting plane methods, which are known to be slow. Bundle methods add a regularization layer over the classical Kelley's algorithm, and have been shown to greatly improve its numerical efficiency. That is why there have been several contributions made toward regularizing the SDDP algorithm, with some partial successes. Usually, regularization is obtained by adding a term penalizing the distance to the last iterate. Unfortunately, for MSPs, the last iterate should be the stochastic process of the state, that is its value over the whole tree, and not the realization on a single scenario, making a direct adaptation intractable.

Another path, known as level-regularization, consists in using an updated upper bound on the objective function. This approach has been adapted to SDDP algorithms in [VAdOS19], where the question of having good upper bounds remained open. It fit the TDFP framework as a modification of the forward operator $\mathcal{A}_t$.

4.3.3 Inner approximation policy

A less explored use of upper bounds is to leverage them to define a policy. Indeed, in SDDP we obtain a collection of lower-approximation of risk-adjusted cost-to-go functions $(\underline{V}_t)$, which defines an admissible policy

$$\pi_t^{\underline{V}_t} : x, \xi \mapsto \mathcal{A}_t(\underline{V}_t)(x, \xi) \ . \tag{4.24}$$

If we are updating upper approximations $(\overline{V}_t)$, we might alternatively use a policy $\pi_t^{\overline{V}_t}$ if it is well-defined. It might not be if $\operatorname{dom}(\overline{V}_t) \subsetneq \operatorname{dom}(V_t)$, which is often the case for the upper bound generated

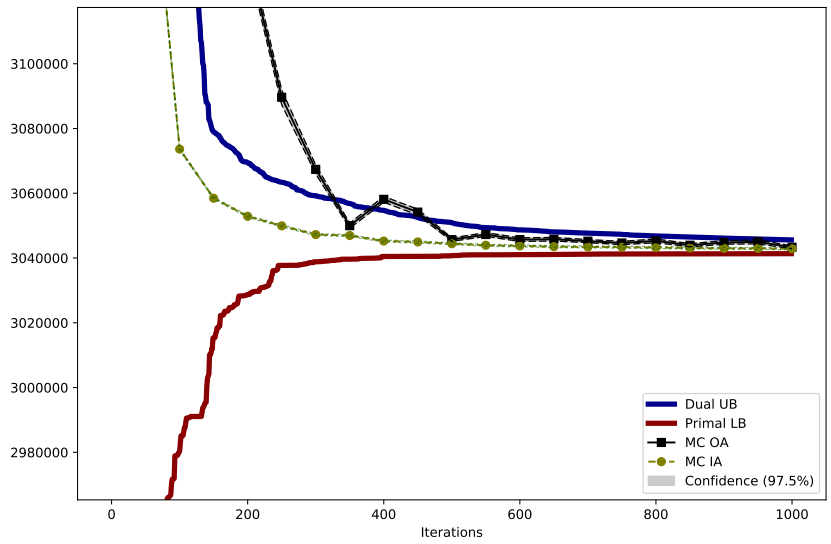


Figure 4.1: Convergence of the exact lower and upper bounds, and Monte Carlo estimation of the cost induced by the outer approximation (MC-OA) and inner approximation (MC-IA).

from the primal backward recursion (4.3). Indeed, in this case, the domain of the upper bound function is the convex hull of the trial points, and a relatively complete recourse assumption does not guarantee that we can get there. However, if we know that the value function is L_t-Lipschitz, we can use instead the Lipschitz-regularized upper bound $\overline{V}_t^L$ (see (4.4)).

This idea is explored in [VL10], Section 4. In particular, we can adapt the proof to show that $\overline{V}_t^L(x)$ is not only an upper bound on the optimal value of the problem $V_t(x)$, but also on the risk-adjusted cost incurred by the policy $\pi_t^{\overline{V}_t^L}$. Meaning that inner approximation induced strategy can guarantee the quality of strategy in the risk-averse setting. This is in stark contrast with the usual risk-averse SDDP methodology where not only estimating an upper bound to the optimal value of the problem is difficult, but even estimating the risk-adjusted cost incurred by a given strategy is difficult.

Finally, numerical results have shown that inner approximation strategies are well-behaved. Indeed, classical outer approximation based strategies can be non-monotone in iteration, meaning that the quality of a strategy can decrease when adding cuts. It seems that inner approximation based strategies are better behaved in this regard (see Fig. 4.1)

Part II

Applications of optimization under uncertainty

Introduction to Part II

Since the beginning of my academic life, I have had the opportunity to contribute to various industrial collaborations that I would like to acknowledge here. They provide the scientific materials for the second part of this manuscript.

Management of smart-grids

With Michel de Lara, Pierre Carpentier and Jean-Philippe Chancelier we produced a report [VL2] on the optimization methods for the smart grid for the French Energy Council. The report covers various tools for stochastic optimization to tackle the renewable energy and demand uncertainty, which are briefly presented in Chapter 5.

The main idea is to cast smart-grid management problems as large-scale, structured, MSPs. The goal is to present mathematical tools that allow decomposing these large-scale MSPs into smaller more tractable ones. In particular, as part of a long-term collaboration with EdF, we developed a method coupling a spatial decomposition approach, to cut the energy network into small elements, and a time-decomposition approach using the Dynamic Programming tools presented in Part I.

Game for energy markets

Energy markets are very complex, for various reasons, and the subject of a large amount of work.

In collaboration with the Program Gaspard Monge for optimization (PGMO), I have worked on various projects related to energy management and energy markets. Most of the work presented in Part I has been motivated and supported in part by PGMO projects.

In Chapter 6, I present the result of a specific project, carried out with Andy Philpott. The objective is to study the impact of risk aversion on the agents in energy markets. Indeed, in a risk-neutral setting, an equilibrium in a complete market is equivalent to the social optimum problem. We show in [VL8] that it is no longer the case if the agents are risk-averse. Furthermore, we show that risk-aversion, even with strong convexity assumptions, there are multiple equilibria, and standard off-the-shelf tools might select an unstable equilibrium.

Balancing service level, stock and cost in the supply chain

The Ph.D. grant of Étienne de Saint Germain, co-supervised with Frédéric Meunier, was funded by Argon&Co, a management consulting company focusing on supply chain problems. The objective was to provide mathematical models and optimization methods to find a balance between cost, inventory stock and service level. This work is presented, in part, in Chapter 7.

Chapter 5

Decomposition tools and applications in energy management

In this chapter, we present decomposition techniques for large-scale multistage stochastic program (MSP) and a use case in energy management. This work derives from my Ph.D. thesis work and long-term collaboration with P. Carpentier, J-Ph. Chancelier, M. De Lara as well as former student F. Pacaud.

We first give a generic presentation of decomposition methods in Section 5.1 for large-scale MSPs. More details can be found in the technical report [VL2]. Section 5.2 is dedicated to the Dual Approximate Dynamic Programming (DADP) approach, which can be understood as a Lagrangian decomposition method for a relaxed problem. Finally, Section 5.3 presents a hydroelectric valley application of DADP that can be found in [VL7].

For simplicity, and algorithmic reasons, we make in this chapter the finite noise assumption Assumption (FSN), and stagewise independence assumption Assumption (SWI).

5.1 Decomposition methodologies for large-scale multistage problems

We present here a structured version of the risk-neutral MSP Problem (1.9). We aim to underlie how the large-scale problem can be seen as a sum of independent problems linked by coupling constraints. First, § 5.1.1 introduces the setting and notations. Then, § 5.1.2 presents the Dynamic Programming approach as a sequential time-decomposition method. § 5.1.3 use Lagrangian duality theory to decompose the MSP in deterministic problems, ending with a quick presentation of the well-known Progressive-Hedging algorithm of Rockafellar and Wets [Wet89]. Finally, § 5.1.4 also uses Lagrangian duality to decouple the MSP problem. This last decomposition is then improved upon in Section 5.2.

Before diving in we give some notational pointers to ease understanding. Generically speaking, $i \in [I]$ represent a unit, $t \in [T]$ a stage, $s \in S$ a scenario, k an iteration in an algorithm. As usual, $\boldsymbol{x}$ represents a state, $\boldsymbol{u}$ a control and $\boldsymbol{\xi}$ a noise. For example, $\boldsymbol{x}_t^i$ represents the local state of unit i at stage t. When an index is omitted, it means that we consider the collection, *e.g.,* $\boldsymbol{x}_t = (\boldsymbol{x}_t^i)_{i\in[I]}$ and $\boldsymbol{x}^i = (\boldsymbol{x}_t^i)_{t\in[T]}$.

5.1.1 Coupled multistage problem setting

Consider I controlled stochastic dynamic system following a dynamic equation

$$\boldsymbol{x}_0^i \text{ given.} \qquad \boldsymbol{x}_t^i = \mathrm{dyn}_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t), \quad \forall i \in [I], \forall t \in [T].$$

As in Problem (1.9), each system have an instantaneous cost L_t^i, a final cost V_{T+1}^i and have their control $\boldsymbol{u}_t^i$ constrained to be in $\mathcal{U}_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{\xi}_t)$. Further, these dynamical systems are coupled additively by the following constraint

$$\sum_{i=1}^{I} \theta_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t) = 0 \qquad \forall t \in [T],$$

where $\theta_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t)$ can be understood as the output of system i which interact with the other stochastic dynamical systems.

We aim to minimize the expected sum of costs over the I systems resulting in the following extension[1] of Problem (1.9).

$$\begin{aligned}
\underset{(\boldsymbol{u}_t^i)_{t\in[T],i\in[I]}}{\text{Min}} \quad & \mathbb{E}_{\mathbb{P}}\Big[\sum_{i=1}^{I}\sum_{t=1}^{T} L_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t) + V_{T+1}^i(\boldsymbol{x}_T^i)\Big] && && \text{(5.1a)}\\
\text{s.t.} \quad & \boldsymbol{x_t^i} = \mathrm{dyn}_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t), && \forall t\in[T], \forall i\in[I], & \text{(5.1b)}\\
& \boldsymbol{u}_t^i \in \mathcal{U}_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{\xi}_t), && \forall t\in[T], \forall i\in[I], & \text{(5.1c)}\\
& \sum_{i=1}^{I} \theta_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t) = 0, && \forall t\in[T], & \text{(5.1d)}\\
& \boldsymbol{u}_t^i \preccurlyeq \mathcal{A}_t, && \forall t\in[T], \forall i\in[I]. & \text{(5.1e)}
\end{aligned}$$

Assume that the set of scenarios S is finite, each $s \in S$ having probability π^s. Each scenario s defines a sequence of noise $(\xi_t^s)_{t\in[T]}$. We define the *bundle* of scenarios that coincides with s up to time t as

$$n_t^s := \Big\{ s' \in S \quad | \quad \xi_\tau^{s'} = \xi_\tau^{s'}, \quad \forall \tau \in [t] \Big\}. \tag{5.2}$$

Note that, for all scenario $s' \in n_t^s$, we have $n_t^s = n_t^{s'}$. This stochastic structure can thus be represented on a tree $\mathcal{T}$[2], each node n being defined as a bundle of scenarios. We denote $\mathcal{W}$ the set of non-anticipative control, that is satisfying (5.1e), which we can denote

$$\mathcal{W} := \Big\{ \boldsymbol{u} \quad | \quad u_{[t]}^{i,s} = u_{[t]}^{i,s'} \qquad \forall s' \in n_t^s, \quad \forall t \in [T], \quad \forall i \in [I] \Big\}. \tag{5.3}$$

With these notations, we can give a splitted, extended formulation of the above Problem (5.1)

$$\begin{aligned}
\underset{(u_t^i)_{t\in[T],i\in[I]}}{\text{Min}} \quad & \sum_{s\in S} \pi^s \sum_{i=1}^{I}\sum_{t=1}^{T} L_t^i(x_{t-1}^{i,s}, u_t^{i,s}, \xi_t^s) + V_{T+1}^i(x_T^s) && && \text{(5.4a)}\\
\text{s.t.} \quad & x_t^{i,s} = \mathrm{dyn}_t^i(x_{t-1}^{i,s}, u_t^{i,s}, \xi_t^s), && \forall t\in[T], \forall i\in[I], \forall s\in S, & \text{(5.4b)}\\
& u_t^{i,s} \in \mathcal{U}_t^i(x_{t-1}^{i,s}, \xi_t^s), && \forall t\in[T], \forall i\in[I], \forall s\in S, & \text{(5.4c)}\\
& \sum_{i=1}^{I} \theta_t^i(x_{t-1}^{i,s}, u_t^{i,s}, \xi_t^s) = 0, && \forall t\in[T], \forall s\in S, & \text{(5.4d)}\\
& u_{[t]}^{i,s} = u_{[t]}^{i,s}, && \forall s'\in n_t^s, \forall s\in S, \forall t\in[T], \forall i\in[I]. & \text{(5.4e)}
\end{aligned}$$

Note, in particular, that for every $t \in [T]$ and $i \in [I]$, we have one state and control variable per scenario. The non-anticipativity constraint (5.1e) are represented here through the equality constraints (5.4e).

This large, coupled, Problem (5.4) is illustrated through Fig. 5.1, where each node of the cube represents an element of the sum in the objective, and each link a coupling constraint. More precisely, the horizontal lines, parallel to "time" axis, represent the dynamic constraints (5.4b); the horizontal lines, parallel to "uncertainty" axis, represent the non-anticipativity constraints (5.4e) coupling scenarios together; the vertical links, parallel to "unit" axis represent the spatial coupling constraint (5.4d).

The remainder of this section gives an overview of how this large-scale problem can be decomposed into smaller problems that can be more efficiently solved. The key ingredient, except for Dynamic Programming, consists in dualizing the coupling constraint - also known as Lagrangian relaxation - in order

[1] One could also claim that Problem (5.1) is a specific, structured, instance of Problem (1.9).

[2] The construction of the scenario tree presented here slightly differ from the one given in Definition 1.11, but there is a one-to-one correspondence between both description.

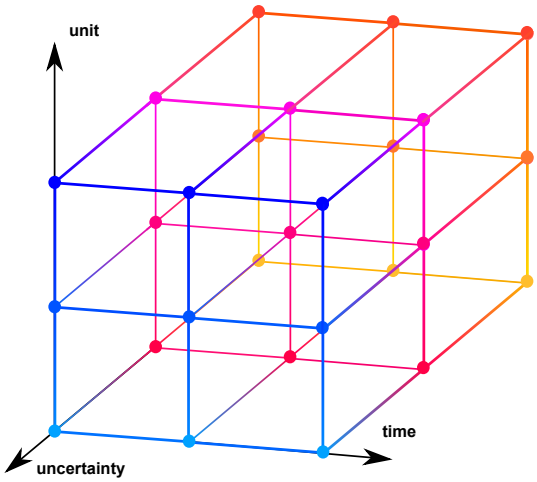
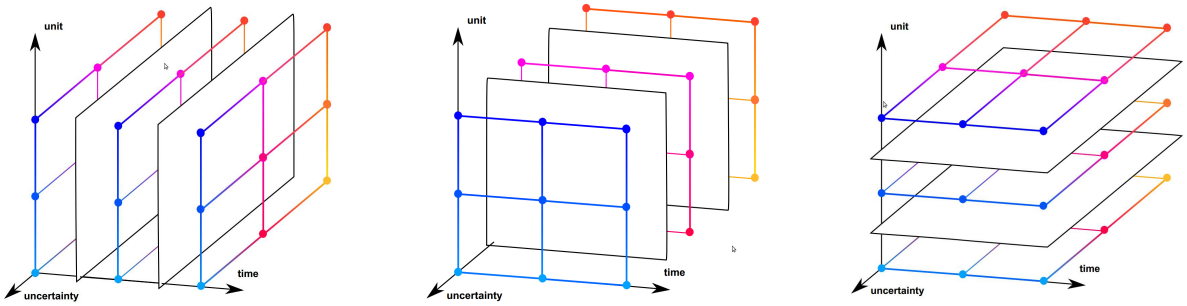


Figure 5.1: Illustration of the coupled large-scale optimization problem



(a) Time decomposition (b) Scenario decomposition (c) Spatial decomposition

Figure 5.2: Decomposition of large-scale multistage problems

to obtain a sum of decoupled problems. As such convexity is a key assumption. These decompositions are illustrated in Fig. 5.2. Note that having links not parallel to the axes, for example, delay between the action on one unit i and its impact on other units, would render the decomposition approaches presented here more difficult.

5.1.2 Time-decomposition: Dynamic Programming

As seen in § 1.2.3, and largely exploited throughout Part I the Dynamic Programming principle ensures that, under a stage-wise independence assumption, we can reduce the T stage problem into T (parametrized) one-stage problem. Thus, we can see dynamic programming as a time decomposition of the large-scale MSP problem.

More precisely, solving Problem (5.4) by Dynamic Programming can be done through the use of a

coupled Bellman operator $\mathcal{B}_t$ defined as

$$\hat{\mathcal{B}}_t(R) : (x, \xi_t) \mapsto \underset{(u_t^i)_{i\in[I]}}{\text{Min}} \quad \sum_{i=1}^{I} L_t^i(x_{t-1}, u_t, \xi_t) \; + \; R\Big((x_{t+1}^i)_{i\in[I]}\Big) \tag{5.5a}$$

$$\text{s.t.} \quad x_t^i = \text{dyn}_t^i(x_{t-1}^i, u_t^i, \xi_t), \qquad \forall i \in [I],$$

$$u_t^i \in \mathcal{U}_t^i(x_{t-1}, \xi_t), \qquad \forall i \in [I],$$

$$\sum_{i=1}^{I} \theta_t^i(x_{t-1}^i, u_t^i, \xi_t) = 0,$$

and

$$\mathcal{B}_t(R) : x \mapsto \sum_{\xi\in\text{supp}(\boldsymbol{\xi}_t)} \mathbb{P}(\boldsymbol{\xi}_t = \xi)\; \hat{\mathcal{B}}_t(R)(x, \xi). \tag{5.5b}$$

Note that, here, the stagewise independence assumption is required for the sequence of noise $(\boldsymbol{\xi}_t)_{t\in[T]}$, that is across time. If the noise $\boldsymbol{\xi}_t$ is a vector of noises $(\boldsymbol{\xi}_t^i)_{i\in[I]}$, then dependence between them would not hinder Dynamic Programming. It could even help, by reducing the support of $\boldsymbol{\xi}_t$. Due to the curse of dimensionality (see Remark 1.15), this approach is numerically intractable if the number of unit I is larger than 4 or 5, even if each unit is unidimensional.

Further, we define a coupled forward Bellman operator $\mathcal{F}_t(R)$ that return, for each x_{t-1}, ξ_t the optimal x_t of Problem (5.5a). Thus, any approximation $\widetilde{V}_t$ of the true coupled value function V_t defines an admissible policy for the coupled Problem (5.1). Given $(\widetilde{V}_t)_{t\in[T]}$ simulating the corresponding policy along a scenario consists thus in solving T deterministic, one-stage, coupled problem. This is used in Sections 5.2 and 5.3 to reconstruct admissible policy from the approximation made.

5.1.3 Scenario-decomposition: Progressive Hedging

Scenario decomposition approaches consist in dualizing the non-anticipativity constraint, to see the stochastic Problem (5.4) as a sum of spatially coupled, T-stage, deterministic problems (one per scenario).

Remark 5.1 (Stagewise independence assumption)**.** *Scenario decomposition, either vanilla Lagrangian decomposition, or Progressive-Hedging, does not require nor make use of the stagewise independence assumption. Consequently, it does not use the dynamic programming principle to compress information, and we need to consider all scenarios in S. In the end, these methods are only used for small horizon T.*

The first step in this approach is to rewrite the non-anticipativity constraints (5.4e) as

$$u_t^{i,s} = \frac{1}{\pi_{n_t^s}} \sum_{s'\in n_t^s} \pi_{s'} u_t^i(s') \qquad \forall s \in S, \forall t \in [T], \forall i \in [I], \tag{5.6}$$

where $\pi_{n_t^s} \coloneqq \sum_{s'\in n_t^s} \pi_{s'}$ is the probability of node n_t^s. Interestingly, we can then show that

$$\mathcal{M} \coloneqq \mathcal{W}^\perp = \Big\{\lambda \quad | \sum_{s'\in n_t^s} \pi_{s'} \lambda_t^{s'} = 0 \quad \forall t \; \in [T], \quad \forall s \in S\Big\}. \tag{5.7}$$

Finally, for notational sobriety we define, for $s \in S$, the state trajectory

$$x_0^{i,s}(u) = x_0, \qquad x_t^{i,s}(u) = \text{dyn}_t^i(x_{t-1}^{i,s}(u), u_t, \xi_t^s), \qquad \forall i \in \; [T], \forall i \in [I], \tag{5.8}$$

and the scenario-cost function

$$J^{i,s}(u) = \sum_{t=1}^{T} \left\{L_t^i(x_{t-1}^{i,s}(u), u_t, \xi_t^s) + \mathbb{I}_{u_t\in\mathcal{U}_t^i(x_{t-1}^{i,s}, \xi_t^s)}\right\} + V_{T+1}(x_T^{i,s}(u)), \tag{5.9a}$$

$$J^s(u) = \sum_{i=1}^{I} J^{i,s}(u). \tag{5.9b}$$

Combining all of this, and dualizing the non-anticipativity constraint we get

$$\max_{\lambda_n^i \in \mathcal{M}} \quad \sum_{s\in S} \pi^s \min_{(u_t^i)_{t\in[T],i\in[I]}} \quad J^s(u) + \sum_{i\in[I]} \sum_{t\in[T]} \lambda_t^{i,s} u_t^{i,s} \tag{5.10a}$$

$$\text{s.t.} \quad \sum_{i=1}^{I} \theta_t^i(x_{t-1}^{i,s}, u_t^i, \xi_t^s) = 0, \qquad \forall t \in [T]. \tag{5.10b}$$

Let $\mathcal{D}(\lambda, s)$ be the value of the inner minimization problem. By weak duality, for any $\lambda \in \mathcal{M}$, $\sum_{s\in S} \pi^s \mathcal{D}(\lambda, s)$ is a lower bound to the value of Problem (5.1). This lower bound is obtained by solving $|S|$ deterministic problems, but can also be estimated by Monte Carlo approaches. Further, under convexity assumptions, and constraint qualifications, we can show that this lower bound is tight. It is then natural to consider a dual ascent algorithm as presented in Algorithm 6. The name comes from the fact that the multiplier update in Line 6 can be interpreted, through envelope theorems, as a (sub)-gradient step for the maximization problem in Problem (5.10).

Data: Scenario tree, information price process $\lambda^{(0)} \in \mathcal{M}$, parameter $r > 0$
1 **for** $k \in \mathbb{N}$ **do**
2 **for** $s \in S$ **do**
3 $v^{(k+1),s} \coloneqq \arg\min_v \quad J^s(v) + \langle \lambda^{(k)}, v \rangle$
4 **for** $s \in S, t \in [T]$ **do**
5 $u_t^{(k+1),s} \coloneqq \frac{1}{\pi_{n_t^s}} \sum_{s' \in n_t^s} \pi_{s'} v^{(k+1),s'}$
6 $\lambda^{(k+1),s} = \lambda^{(k),s} + r(v^{(k+1),s} - u^{(k+1),s})$

Algorithm 6: Dual ascent for scenario decomposition

where $\langle \lambda^{(k)}, v \rangle = \sum_{s\in S} \pi^s \sum_{t=1}^{T} (\lambda_t^{(k),s})^\top v_t^s$ such that the scenario problem reads

$$\min_{(u_t^{i,s})_{t\in[T],i\in[I]}} \quad \sum_{i=1}^{I} \sum_{t=1}^{T} \left\{ L_t^i(x_{t-1}^{i,s}, u_t^{i,s}, \xi_t^s) + (\lambda_t^{(k),s,i})^\top u_t^{i,s} \right\} + V_{T+1}^i(x_T) \tag{5.11a}$$

$$\text{s.t.} \quad x_t^{i,s} = \mathrm{dyn}_t^i(x_{t-1}^{i,s}, u_t^{i,s}, \xi_t^s), \qquad \forall t \in [T], \forall i \in [I], \tag{5.11b}$$

$$u_t^{i,s} \in \mathcal{U}_t^i(x_{t-1}^{i,s}, \xi_t^s), \qquad \forall t \in [T], \forall i \in [I], \tag{5.11c}$$

$$\sum_{i=1}^{I} \theta_t^i(x_{t-1}^{i,s}, {}^{i,s}_{t}, \xi_t^s) = 0, \qquad \forall t \in [T]. \tag{5.11d}$$

This approach has been developed and popularized, in a more efficient version using augmented Lagrangian duality, under the name Progressive Hedging [Wet89], with heuristic extensions to mixed-integer problems [WW11]. It can be interpreted as a fixed-point method over a splitting operator [Rus97], from which the convergence analysis was derived, and is still extended, for example to randomized version [BLG+20]. The Progressive Hedging algorithm is presented in Algorithm 7.

Data: Scenario tree, decision process $u^{(0)} \in \mathcal{W}$, information price process $\lambda^{(0)} \in \mathcal{M}$, parameter $r > 0$
1 **for** $k \in \mathbb{N}$ **do**
2 **for** $s \in S$ **do**
3 $v^{(k+1),s} \coloneqq \arg\min_v \quad J^s(v) + \frac{r}{2}\|v - u^{(k),s} + \frac{1}{r}\lambda^{(k),s}\|^2$
4 **for** $s \in S, t \in [T]$ **do**
5 $u_t^{(k+1),s} \coloneqq \frac{1}{\pi_{n_t^s}} \sum_{s' \in n_t^s} \pi_{s'} v^{(k+1),s'}$
6 $\lambda^{(k+1),s} = \lambda^{(k),s} + r(v^{(k+1),s} - u^{(k+1),s})$

Algorithm 7: Progressive Hedging algorithm

Let's end this section with a few comments on the Progressive Hedging algorithm, that decompose the MSP into multiple deterministic problems (in Line 3), and then coordinates them (in Line 6):

i) if the MSP problem is convex, then the scenario problem is strongly convex and $v_t^{(k),s}$ uniquely defined;

ii) the scenario problem in Line 3 can be written as, up to additive constant,

$$\min_v \qquad J^s(v) + \langle \lambda, v \rangle + \frac{r}{2}\|v - u^{(k),s}\|^2,$$

which yields a quadratic regularization interpretation of the Progressive Hedging approach over the vanilla Lagrangian decomposition presented above;

iii) the scenario problem is a deterministic large-scale problem, and, if needed, standard deterministic decomposition methods could be applied to it;

iv) the projection step Line 5 is used to determine a non-anticipative solution from the $|n_t^s|$ available at every step, this step is difficult to extend to the non-convex (especially integer) case;

v) to reduce the number of deterministic problems, one can consider extensive formulation on subtree instead of on a single scenario.

In the end, the Progressive Hedging algorithm is an efficient tool to decompose an MSP problem into deterministic problems and leverage available tools. One of its main limits is that we have to solve $|S|$ deterministic problems at each iteration. This usually restricts the use of Progressive Hedging to problems of limited horizon T.

5.1.4 Spatial-decomposition

The third decomposition axis consists in separating the coupled large-scale MSP, into I independent MSPs. To this end, we dualize the spatial coupling constraint (5.1d), to obtain the following dual problem

$$\begin{aligned}
&\operatorname*{Max}_{\boldsymbol{\lambda}} && \sum_{i=1}^{I} V^i(\boldsymbol{\lambda}) && && \text{(5.12a)}\\
V^i(\boldsymbol{\lambda}) = &\operatorname*{Min}_{\boldsymbol{u}^i} && \mathbb{E}_{\mathbb{P}}\left[\sum_{t=1}^{T}\left\{L_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t) + \boldsymbol{\lambda}_t \theta_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t)\right\} + V_{T+1}^i(\boldsymbol{x}_T^i)\right] && && \text{(5.12b)}\\
&\text{s.t.} && \boldsymbol{x_t^i} = \mathrm{dyn}_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t), && \forall t \in [T],\\
& && \boldsymbol{u}_t^i \in \mathcal{U}_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{\xi}_t), && \forall t \in [T],\\
& && \boldsymbol{u}_t^i \preccurlyeq \mathcal{A}_t && \forall t \in [T].
\end{aligned}$$

As before, for any multiplier process $\boldsymbol{\lambda}$, $\sum_{i=1}^{I} V^i(\boldsymbol{\lambda})$ is a lower bound to the value of the coupled MSP Problem (5.1). If the coupled problem is convex, and under constraint qualification assumptions, we can show that there exists an optimal multiplier $\boldsymbol{\lambda}$ such that this lower bound is tight. We then get a dual-ascent algorithm.

Data: Spatial price process $\lambda^{(0)}$, parameter $r > 0$
1 **for** $k \in \mathbb{N}$ **do**
2 **for** $i \in [I]$ **do**
3 Solve $V^i(\boldsymbol{\lambda}^{(k)})$ for state and control trajectories $(\boldsymbol{u}_t^{(k),i}, \boldsymbol{x}_t^{(k),i})_{t\in[T]}$
4 Define the slack process $\boldsymbol{\Delta}_t^{(k)} \coloneqq \sum_{i\in[I]} \theta_t^i(\boldsymbol{x}_{t-1}^{(k),i}, \boldsymbol{u}_t^{(k),i}, \boldsymbol{\xi}_t)$
5 Update spatial price process $\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + r\boldsymbol{\Delta}$

Algorithm 8: Dual ascent for spatial decomposition algorithm

In this approach, $\boldsymbol{\lambda}$ is a stochastic process, such that λ_t^s is a vector of dimension the image of θ_t^i. Without loss of generality, we can consider non-anticipative $\boldsymbol{\lambda}$. Consequently, if the horizon T is small enough, we can decompose the large-scale MSP into I MSPs, solve them independently and coordinate them through the price process $\boldsymbol{\lambda}$.

However, if we have a stagewise independence assumption for the large-scale MSPs, thus enabling Dynamic Programming approaches, this is no longer possible on the decomposed problem. Indeed, when solving $V^i(\boldsymbol{\lambda}^{(k)})$, we have two different noises $(\boldsymbol{\xi}_t)_{t\in[T]}$, and $(\boldsymbol{\lambda}^{(k)})_{t\in[T]}$, and there is no reason for spatial multiplier $\boldsymbol{\lambda}^{(k)}$ to be stagewise independent, which would allow dynamic programming methods. The next section offers a possible solution.

5.2 Dual approximate dynamic programming (DADP)

The idea of the Dual Approximate Dynamic Programming, originated in [BCG10], then improved during various Ph.D. theses, including my own, is to combine spatial and time-decomposition methods. More precisely we want to use a spatial decomposition method as presented in § 5.1.4, in a way such that the subproblems can be solved through Dynamic Programming.

5.2.1 DADP principle

We introduce a non-anticipative information process $(\boldsymbol{\zeta}_t)_{t\in[T]}$, such that $\boldsymbol{\zeta}_t$ is measurable with respect to the noises up to time t, that is $\mathcal{A}_t = \sigma(\boldsymbol{\xi}_{[t]})$. We then consider a relaxed version of the coupled MSP (5.1), where the almost-sure coupling spatial constraint is relaxed in a conditional expectation constraint, *i.e.,*

$$\begin{aligned}
\underset{(\boldsymbol{u}_t^i)_{t\in[T],i\in[I]}}{\text{Min}} \quad & \mathbb{E}_{\mathbb{P}}\Big[\sum_{i=1}^{I}\sum_{t=1}^{T} L_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t) + V_{T+1}^i(\boldsymbol{x}_T^i)\Big] && && \text{(5.13a)}\\
\text{s.t.} \quad & \boldsymbol{x_t^i} = \mathrm{dyn}_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t) && \forall t\in[T], \forall i\in[I], && \text{(5.13b)}\\
& \boldsymbol{u}_t^i \in \mathcal{U}_t^i(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t) && \forall t\in[T], \forall i\in[I], && \text{(5.13c)}\\
& \mathbb{E}\Big[\sum_{i=1}^{I} \theta_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t) \;\Big|\; \boldsymbol{\zeta}_t\Big] = 0 && \forall t\in[T], && \text{(5.13d)}\\
& \boldsymbol{u}_t^i \preccurlyeq \mathcal{A}_t && \forall t\in[T], \forall i\in[I]. && \text{(5.13e)}
\end{aligned}$$

Dualizing the relaxed coupling constraint (5.13d) we get the following problem

$$\begin{aligned}
\underset{(\boldsymbol{\mu}_t\preccurlyeq\boldsymbol{\zeta}_t)_{t\in[T]}}{\text{Max}} \quad & \sum_{i=1}^{I} V^i(\boldsymbol{\mu}), && && \text{(5.14a)}\\
V^i(\boldsymbol{\mu};\boldsymbol{\zeta}) = \underset{\boldsymbol{u}^i}{\text{Min}} \quad & \mathbb{E}\Big[\sum_{t=1}^{T}\Big\{L_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t) + \boldsymbol{\mu}_t\theta_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t)\Big\} + V_{T+1}^i(\boldsymbol{x}_T^i)\Big] && && \text{(5.14b)}\\
\text{s.t.} \quad & \boldsymbol{x_t^i} = \mathrm{dyn}_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t), && \forall t\in[T], &&\\
& \boldsymbol{u}_t^i \in \mathcal{U}_t^i(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t), && \forall t\in[T], &&\\
& \boldsymbol{u}_t^i \preccurlyeq \mathcal{A}_t, && \forall t\in[T], &&
\end{aligned}$$

where the spatial price $\boldsymbol{\mu}_t$ can be chosen, without loss of generality, as measurable with respect to $\boldsymbol{\zeta}_t$. By weak duality, the value of this dual problem is a lower bound of the value of the relaxed Problem (5.13), which is itself a lower bound of the value of the coupled Problem (5.1). With convexity and constraint qualification assumptions, we have equality between the value of Problem (5.14) and Problem (5.13).

Remark 5.2 (Interpretations of DADP)**.** *The Dual Approximate Dynamic Programming consists in decomposing spatially the coupled problem in a way such that each subproblem can be solved by dynamic*

programming, which requires an approximation. There is three different interpretation of this approximation:

1. *As presented here, it can be seen as a relaxation of the almost sure constraint in the primal problem. Once relaxed, we can dualize the coupling constraint and have subproblems solvable by dynamic programming.*
2. *It can equivalently be seen as a decision rule in dual. Indeed, relaxing primal constraint* (5.1d) *is equivalent to asking the dual multiplier process $\boldsymbol{\lambda}$ in Problem* (5.12) *to be progressively measurable with respect to $(\boldsymbol{\zeta}_t)_{t\in[T]}$.*
3. *Finally, it can be seen as an approximation method in the dual, where instead of using the dual stochastic process $\boldsymbol{\lambda}$ we use its conditional expectation $\boldsymbol{\mu}_t = \mathbb{E}\big[\boldsymbol{\lambda}_t \bigm| \boldsymbol{\zeta}_t\big]$.*

5.2.2 Algorithmic implementation of DADP

To go further, we assume that the information process follows a dynamic equation

$$\boldsymbol{\zeta}_t = h_t(\boldsymbol{\zeta}_{t-1}, \boldsymbol{\xi}_t), \qquad \forall t \in [T], \tag{5.15}$$

where $\boldsymbol{\zeta}_0$ is arbitrarily set.

We can then solve each subproblem $V^i(\boldsymbol{\mu}, \boldsymbol{\zeta})$ (see (5.12b)) through dynamic programming using the local physical state and the information process, that is the couple $(\boldsymbol{x}_t^i, \boldsymbol{\zeta}_t)$ as a state. More precisely, we have the following local backward Bellman operator

$$\begin{aligned}
\hat{\mathcal{B}}_t^i(R;\mu) : (x_{t-1}^i, \zeta_{t-1}; \xi_t) \mapsto \quad & \min_{u_t^i} \quad L_t^i(x_{t-1}^i) + \mu(\zeta_t)\theta_t^i(x_{t-1}^i, u_t^i, \xi_t) + R(x_t^i, \zeta_t) && \text{(5.16a)}\\
& \text{s.t.} \quad x_t^i = \mathrm{dyn}_t^i(x_{t-1}^i, u_t^i, \xi_t),\\
& \qquad\quad \zeta_t = h_t(\zeta_{t-1}, \xi_t),\\
\mathcal{B}_t^i(R;\mu) : (x_{t-1}^i, \zeta_{t-1}) = \quad & \mathbb{E}\big[\hat{\mathcal{B}}_t^i(R)(x_{t-1}^i, \zeta_{t-1}; \boldsymbol{\xi}_t)\big]. && \text{(5.16b)}
\end{aligned}$$

We have, accordingly, local forward Bellman operators, which define a local policy, that is a function of time t, local state x_t^i, information process ζ_t and noise ξ_t that return a local control u_t^i.

The DADP scheme is thus illustrated in Fig. 5.3

Remark 5.3 (Recovering an admissible strategy)**.** *When using the DADP algorithm, we obtain a lower bound on the original MSP* (5.1)*. We also obtain, from subproblems resolution, a local strategy that is a function of the local physical state $\boldsymbol{x}_t^i$ and the information process $\boldsymbol{\zeta}_t$. With strong convexity and constraint qualification conditions, we can guarantee that the multiplier process $\boldsymbol{\mu}^{(k)}$ converges. Unfortunately, even at convergence, the local strategy only satisfies the relaxed coupling constraint.*

Consequently, as in TFDP algorithm (see Chapter 3), we use a forward Bellman operator associated with the coupled Problem (5.1) *to obtain an admissible solution from an approximate value function. As the cost of the coupled problem is the sum of local cost, we use, for each $t \in [T]$, the approximate coupled value function*[1]

$$\widetilde{V}_t = \sum_{i=1}^{T} V_t^i.$$

Remark 5.4 (Market interpretation)**.** *The spatial Lagrangian decomposition presented in § 5.1.4 have a market interpretation: we consider that the satisfaction of constraint* (5.1d) *is ensured through a market, where the multiplier process $(\boldsymbol{\lambda}_t)_{t\in[T]}$ is the stochastic clearing price.*

The DADP approach can be interpreted as ensuring that the price at time t $\boldsymbol{\mu}_t$ is a function of the information process $\boldsymbol{\zeta}_t$ instead of a function of past noises $\boldsymbol{\xi}_{[t]}$. This constraint on the price process can either be interpreted as the fact that the restricted clearing price $\boldsymbol{\mu}_t$ is the conditional expectation of the true clearing price $\boldsymbol{\lambda}_t$, i.e., *$\boldsymbol{\mu}_t = \mathbb{E}\big[\boldsymbol{\lambda}_t \bigm| \boldsymbol{\zeta}_t\big]$. This restricted clearing price ensures that the relaxed coupling constraint* (5.13d) *is satisfied.*

[1] Technically we have to compute the information process and make straightforward modification to the equation.

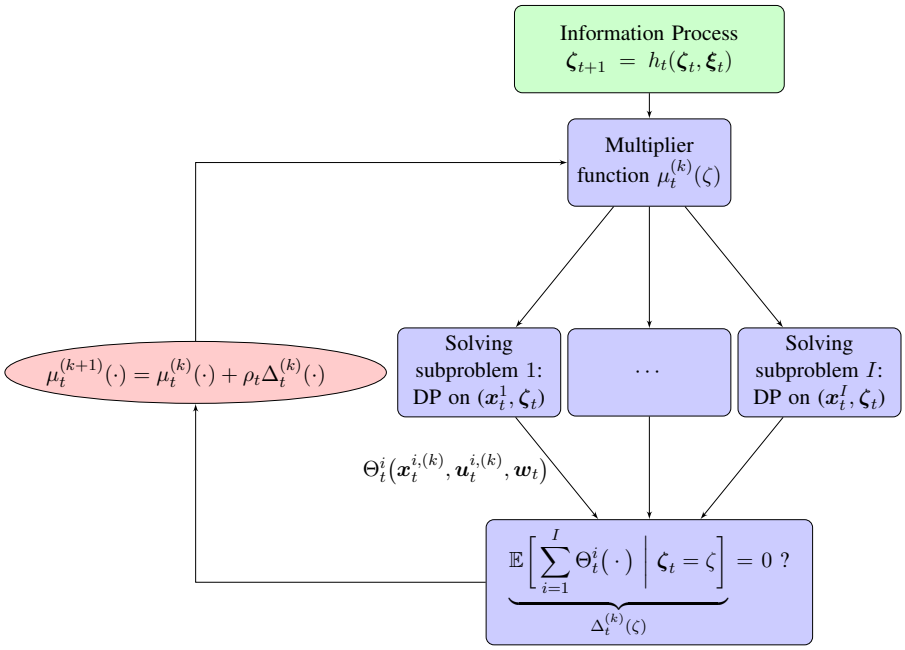


Figure 5.3: DADP flowchart.

5.2.3 Consistency theory

We have seen that the DADP algorithm provides, under strong convexity and constraint qualification assumptions, an optimal solution to Problem (5.13), which is a relaxation of the coupled Problem (5.1). This relaxation consists in replacing almost sure constraints $\theta(z) = 0$ into conditional expectation constraints $\mathbb{E}\big[\theta(z) \,\big|\, \zeta\big] = 0$. The question of the consistency of this approximation is to determine under which condition the relaxed problem converges toward the original problem. Under the finite noise assumption considered in this chapter, the answer is simple, as adding more information end in recovering the full almost sure constraint. In this section, we drop the finite noise assumption to provide a consistency result based on the theory of epi-convergence. Details can be found in [VL9].

Setting

We consider a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, a sequence of σ-algebras $(\mathcal{A}_n)_{n\in\mathbb{N}}$ and a topological space of controls $\mathcal{U}$. Let $\mathcal{V}$ be the space of random variables with value in a Banach $\mathbb{V}$ with finite moment of order $p \in [1, \infty)$, denoted $\mathcal{V} = L^p(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{V})$.

We consider now a stochastic optimization problem

$$\min_{u\in\mathcal{U}} \quad J(u)\ , \tag{5.17a}$$

$$s.t. \quad \Theta(u) \in \mathcal{C}\ , \tag{5.17b}$$

with J mapping $\mathcal{U}$ into $\mathbb{R} \cup \{+\infty\}$, and Θ mapping $\mathcal{U}$ into $\mathcal{V}$. We assume that $\mathcal{C} \subset \mathcal{V}$ is a subset of $\mathcal{V}$, and that $\mathbb{V}$ is a separable Banach space with separable dual.

To give an example of cost operator, assume that $\mathcal{U} \subset L^1\big(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{U}\big)$, where $\mathbb{U}$ is a Banach space. The usual choice for the objective function is the expected cost $J(u) := \mathbb{E}\big[j(u)\big]$, for a suitable cost function $j : \mathbb{U} \to \mathbb{R}$. Other choices could be risk measures like Average-Value-at-Risk, worst-case or robust approaches. The constraint operator Θ cover various cases, for example

- almost sure constraint: $\Theta\big(u\big)(\omega) := \theta\big(u(\omega)\big)$, where θ maps $\mathbb{U}$ into $\mathbb{V}$ and $\theta\big(u\big) \in C$ is realized almost surely, where C is a closed convex set;
- measurability constraint: $\Theta\big(u\big) := \mathbb{E}\big[u \,\big|\, \mathcal{B}\big] - u$, with $\mathcal{C} = \{0\}$, expresses that u is measurable with respect to the σ-algebra $\mathcal{B}$, that is, $\mathbb{E}\big[u \,\big|\, \mathcal{B}\big] = u$;

- risk constraint: $\Theta\big(\boldsymbol{u}\big) := \rho(\boldsymbol{u}) - a$, where ρ is a conditional risk measure, and $\mathcal{C}$ is the cone of negative random variables.

We further assume that the set of constraint C is such that for all $n \in \mathbb{N}$ and all $\boldsymbol{v} \in C$, $\mathbb{E}[\boldsymbol{v} \mid \mathcal{A}_n] \in C$.

We now consider the following relaxation of Problem (5.17)

$$\min_{\boldsymbol{u}\in\mathcal{U}} \quad J(\boldsymbol{u}) \ , \tag{5.18a}$$

$$s.t. \quad \mathbb{E}\big[\Theta(\boldsymbol{u}) \ \big| \ \mathcal{A}_n\big] \in \mathcal{C} \ . \tag{5.18b}$$

We denote $\mathcal{U}^{ad}$ the set of admissible controls of Problem (5.17) and $\mathcal{U}_n^{ad}$ the corresponding set of admissible controls of Problem (5.18). We further define $\widetilde{J}(\boldsymbol{u}) := J(\boldsymbol{u}) + \mathbb{I}_{\mathcal{U}^{ad}}(\boldsymbol{u})$, and

Epi-convergence result

In this section, we show the epi-convergence of the sequence of approximated cost functions $(\widetilde{J}_n)_{n\in\mathbb{N}}$ towards J. For more details and properties of epi-convergence, see Rockafellar-Wets [RW98] in finite dimension, and Attouch [Att84] for infinite dimension.

We start with some recall on the Kudo-convergence of σ-algebras. Let $\mathcal{A}$ be a σ-algebra and $(\mathcal{A}_n)_{n\in\mathbb{N}}$ a sequence of subfields of $\mathcal{A}$ (not necessarily finite nor a filtration). It is said that the sequence $(\mathcal{A}_n)_{n\in\mathbb{N}}$ Kudo-converges toward the σ-algebra $\mathcal{A}_\infty$, and denoted $\mathcal{A}_n \to \mathcal{A}_\infty$, if for each set $F \in \mathcal{A}$, $\Big(\mathbb{E}\big[\mathbb{1}_F \ \big| \ \mathcal{A}_n\big]\Big)_{n\in\mathbb{N}}$ converges in probability toward $\mathbb{E}\big[\mathbb{1}_F \ \big| \ \mathcal{A}_\infty\big]$.

Extending a result of [Kud74, Pic98] we show in [VL9] the Kudo-convergence of $(\mathcal{A}_n)_{n\in\mathbb{N}}$ to $\mathcal{A}_\infty$ is equivalent to the convergence in L^p (strong or weak) of $\mathbb{E}\big[\boldsymbol{X} \ \big| \ \mathcal{A}_n\big]$ toward $\mathbb{E}\big[\boldsymbol{X} \ \big| \ \mathcal{A}_\infty\big]$ for any $\boldsymbol{X} \in L^p$. We further show that both the filtration and the random variable can be converging sequences. More precisely, if $\mathcal{A}_n \to \mathcal{A}_\infty$, and $\boldsymbol{X}_n \to_{L^p} \boldsymbol{X}$ (resp. $\boldsymbol{X}_n \rightharpoonup_{L^p} \boldsymbol{X}$) then $\mathbb{E}\big[\boldsymbol{X}_n \ \big| \ \mathcal{A}_n\big] \to_{L^p} \mathbb{E}\big[\boldsymbol{X} \ \big| \ \mathcal{A}_\infty\big]$ (resp. $\mathbb{E}\big[\boldsymbol{X}_n \ \big| \ \mathcal{A}_n\big] \rightharpoonup_{L^p} \mathbb{E}\big[\boldsymbol{X} \ \big| \ \mathcal{A}_\infty\big]$).

With this result we derive the following convergence theorem where τ denotes the topology of $\mathcal{U}$.

Theorem 5.5. *Let $\mathcal{V} = L^pL^p(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{V})$ be endowed with the strong or weak topology. If the two mappings Θ and J are continuous, and if $(\mathcal{A}_n)_{n\in\mathbb{N}}$ Kudo-converges toward $\mathcal{A}$, then $(\widetilde{J}_n)_{n\in\mathbb{N}}$ epiconverges toward $\widetilde{J}$.*

The direct consequence of this epi-convergence result is that the sequence of Problems (5.18) approximates Problem (5.17) in the following sense. If $(\boldsymbol{u}_n)_{n\in\mathbb{N}}$ is a sequence of controls such that for all $n \in \mathbb{N}$,

$$\widetilde{J}_n(\boldsymbol{u}_n) < \inf_{\boldsymbol{u}\in\mathcal{U}} \widetilde{J}_n(\boldsymbol{u}) + \varepsilon_n, \text{ where } \lim_n \varepsilon_n = 0 \ ,$$

then, for every converging sub-sequence $(\boldsymbol{u}_{n_k})_{k\in\mathbb{N}}$, we have

$$\widetilde{J}\big(\lim_k \boldsymbol{u}_{n_k}\big) = \min_{\boldsymbol{u}\in\mathcal{U}} \widetilde{J}(\boldsymbol{u}) = \lim_k \widetilde{J}_{n_k}\big(\boldsymbol{u}_{n_k}\big) \ .$$

Moreover if $\big(\mathcal{A}_n\big)_{n\in\mathbb{N}}$ is a filtration, then the convergences are monotonous in the sense that the optimal value is non-decreasing in n.

The result is extended and adapted to the multistage case, by having multiple sequences of σ-algebra Kudo-converging.

Example of continuous operators

The epi-convergence result of Theorem 5.5 is based on the continuity of the objective and constraints *operators*. This continuity relies on the choice of operator and the topology used for the control space $\mathcal{U}$.

Using the following technical lemma[1] we obtain examples of continuous operators.

Lemma 5.6. *Let $\Theta : E \to F$, where $(E, \tau_{\mathbb{P}})$ is a space of random variables endowed with the topology of convergence in probability, and (F, τ) is a topological space. Assume that Θ is such that if $(\boldsymbol{u}_n)_{n\in\mathbb{N}}$ converges almost surely toward $\boldsymbol{u}$, then $\Theta(\boldsymbol{u}_n) \to_\tau \Theta(\boldsymbol{u})$. Then Θ is a continuous operator from $(E, \tau_{\mathbb{P}})$ into (F, τ).*

For the objective operator, we can choose a bounded cost function[2] j and a lower-semicontinuous convex risk measure ρ. More precisely, if $\mathcal{U}$ is endowed with the topology of convergence in probability, and $J(\boldsymbol{u}) := \rho\big(j(\boldsymbol{u})\big)$, where $j : \mathbb{U} \to \mathbb{R}$ is continuous and bounded, and ρ a proper lower semi-continuous convex risk measure. Then, $J : \mathcal{U} \to \mathbb{R}$ is continuous.

As for the constraint operator, we can tackle, with the topology of convergence in probability: i) almost sure constraint; ii) risk constraint with coherent risk measures; and iii) measurability constraints (including non-anticipativity constraints) on a dominated subset of $\mathcal{U}$. Measurability constraints are better tackled if $\mathcal{U} = L^{p'}$ is endowed with the strong or weak topology, where $1/p + 1/p' = 1$.

More details and examples can be found in [§3 VL9].

Consistency of DADP

Assume that we have strongly convex costs function L_t^i and final costs V_{T+1}^i, linear dynamic function dyn_t^i, bounded control sets $\mathcal{U}_t^i$ and essentially bounded $\boldsymbol{\xi}_t$. Then, by induction, we can show that the random costs $L_t^i(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t^i, \boldsymbol{\xi}_t)$, as well as the state $\boldsymbol{x}_t^i$ and control $\boldsymbol{u}_t^i$ processes are essentially bounded. In turn, the previous discussion ensures that endowing the coupled control space with the topology of convergence in probability, we can show that the objective and constraint operators are continuous.

Now, assume that we have a sequence of information process $(\boldsymbol{\zeta}_t^{(n)})_{t\in[T],n\in\mathbb{N}}$ such that, for each $t \in [T]$, the generated sequence of σ-algebra $\sigma(\boldsymbol{\zeta}_t^{(n)})$ Kudo-converges toward the true information $\mathcal{A}_t$. Then the value of the relaxed Problem (5.13) converges toward the value of the original Problem (5.1).

This is a theoretical consistency result as in practice we choose only one information process $(\boldsymbol{\zeta}_t)_{t\in[T]}$ of small dimension, to be able to solve each subproblem by Dynamic Programming. The next section details an application of the DADP method.

5.3 A hydroelectric valley application

In this section, we detail a practical use case of DADP for a hydroelectric valley application. The methodology is compared with coupled dynamic programming, and with SDDP.

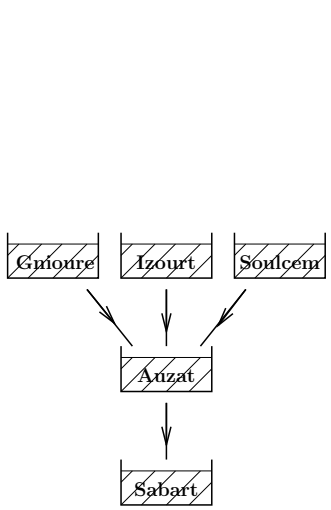
5.3.1 Setting

We consider a hydroelectric valley constituted of I coupled dams as represented in Figure 5.4. We can see each dam as a node, identified by an integer $i \in [I]$ of a directed graph $([I], E)$, where E is the set of edges identified by a couple $(i, j) \in [I]^2$. More precisely, $(i, j) \in E$ if the output of dam i goes into dam j. We denote, for $i \in [I]$, $\delta^+(i)$ (resp. δ^-) the set of edges starting (resp. ending) at i.

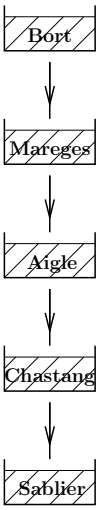
The water released by a dam produces energy which is sold on electricity markets, and then enters the nearest downstream dam. The overall goal of the decision maker is to maximize the profit obtained by selling the produced energy on a market. We consider that the hydro valley manager acts as a price follower, in the sense that the energy prices are independent of the energy produced by the hydro valley.

[1] Even if $(\boldsymbol{u}_n)_{n\in\mathbb{N}}$ converges in probability toward $\boldsymbol{u}$ iff from any sub-sequence of $(\boldsymbol{u}_n)_{n\in\mathbb{N}}$ we can extract a sub-sub-sequence that converges almost surely, this Lemma does not imply the equivalence between convergence almost sure and convergence in probability. Indeed, one cannot endow $\mathcal{U}$ with the "topology of almost sure convergence" as almost sure convergence is not generally induced by a topology.

[2] This assumption can be relaxed by using some dominated convergence theorem.

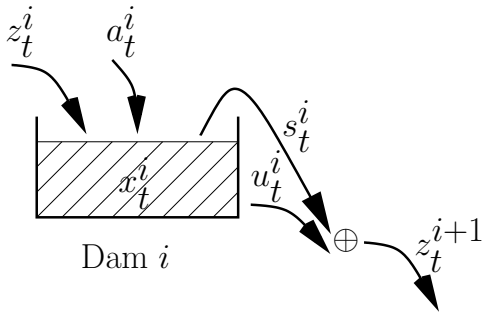


(a) Vicdessos

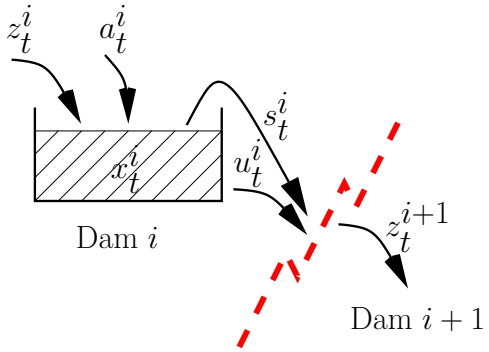


(b) Dordogne

Figure 5.4: Two realistic hydro valleys from south of France.



(a) Compact problem



(b) Duplicated variables

Figure 5.5: In order to use the decomposition scheme we duplicate variables, u_t^i and s_t^i are the local output of dam i while z_t^{i+1} is an inflow variable for dam $i+1$.

The representative variables of dam i at stage t are $\boldsymbol{u}_t^i$ for the released water, $\boldsymbol{x}_t^i$ for the current water volume, $\boldsymbol{a}_t^i$ for the natural water inflow entering dam i, $\boldsymbol{p}_t^i$ for the market value of the water at dam i. The randomness is given by $\boldsymbol{\xi}_t^i = (\boldsymbol{a}_t^i, \boldsymbol{p}_t^i)$. The modeling of a dam takes into account a possible overflow: the spilled water $\boldsymbol{s}_t^i$ does not produce electricity, but enters the next downstream dam. The cost function L_t^i is given as $-\boldsymbol{p}_t^i \boldsymbol{u}_t^i + \varepsilon(\boldsymbol{u}_t^i)^2$ where $\varepsilon > 0$ model a small inefficiency of the turbine and ensures strong convexity. The final cost function V_{T+1}^i is a quadratic penalization around a target value.

In order to introduce a spatial decomposition mechanism, we add a local decision variable $\boldsymbol{z}_t^i$ representing the water obtain by dam i from the previous dams. Consequently, the local dynamic Eq. (5.1b) reads $\boldsymbol{x}_t^i = \boldsymbol{x}_t^i - \boldsymbol{u}_t^i + \boldsymbol{a}_t^i + \sum_{e \in \delta^-(i)} \boldsymbol{z}_t^e - \boldsymbol{s}_t^i$ and the spatial coupling constraints Eq. (5.1d) reads

$$\boldsymbol{z}_t^{(j,i)} = \boldsymbol{u}_t^j + \boldsymbol{s}_t^j, \qquad \forall t \in [T], \forall (i,j) \in E. \tag{5.19}$$

This constraint is the one that is relaxed and dualized in the DADP approach.

5.3.2 DADP implementation

We use the simplest information process $\boldsymbol{\zeta}_t = 0$, which amounts to replacing the almost sure coupling constraint by a constraint in expectation, or, equivalently, looking for deterministic spatial multiplier $(\mu_t^e)_{t \in [T], e \in E}$.

Remark 5.7 (Market interpretation)**.** *Let's illustrate the generic market interpretation of Remark 5.4 on the hydroelectric valley problem at hand.*

The spatial decomposition approach consists in replacing the physical constraint on each link $(i,j) \in E$*, by a market where the total water output of dam* i *is sold to dam* j *at a price* $\boldsymbol{\lambda}_t^{(i,j)}$*. With the strong convexity and constraint qualification of the problem, we know that there exists a clearing price* $(\boldsymbol{\lambda}_t^{(i,j)})_{t\in[T],(i,j)\in E}$ *which is a progressively measurable stochastic process.*

The DADP approach with constant information process $\boldsymbol{\zeta}_t = 0$ *consists in considering a market where the prices* $(\mu_t^{(i,j)})_{t\in[T],(i,j)\in E}$ *are deterministic. In particular, for a given trajectory of prices (one value per link and time-step) managing dam* i *consists in optimizing the sum over time of operational cost* L_t^i*, and value of water sold to the downstream dam and bought to the upstream dams.*

Note that restricting ourselves to deterministic prices means that the market will only satisfy the coupling constraint in expectation, that is $\mathbb{E}\big[\boldsymbol{z}_t^{(i,j)}\big] = \mathbb{E}\big[\boldsymbol{u}_t^i + \boldsymbol{s}_t^i\big]$.

For a given multiplier process μ, the local optimization Problem (5.14b) reads

$$V^i(\mu) = \underset{\boldsymbol{u}^i,\boldsymbol{s}^i,(\boldsymbol{z}^e)_{e\ \in\delta^-(i)}}{\mathrm{Min}} \mathbb{E}\Bigg[\sum_{t=1}^{T}\Big\{ -\boldsymbol{p}_t^i\boldsymbol{u}_t^i + \varepsilon(\boldsymbol{u}_t^i)^2 - \mu_t^{(i,i+)}(\boldsymbol{u}_t^i + \boldsymbol{s}_t^i) + \sum_{i-:(i-,i)\in\delta^-(i)} \mu_t^{(i-,i)}\boldsymbol{z}_t^{(i-,i)}\Big\}$$

$$+ (\boldsymbol{x}_{T+1}^i - \hat{x}_{T+1}^i)^2\Bigg] \tag{5.20a}$$

$$\text{s.t.} \quad \boldsymbol{x_t^i} = \boldsymbol{x_{t-1}^i} - \boldsymbol{u}_t^i - \boldsymbol{s}_t^i + \boldsymbol{a}_t^i + \sum_{i-:(i-,i)\in\delta^-(i)} \boldsymbol{z}_t^{(i-,i)}, \quad \forall t\in[T], \tag{5.20b}$$

$$0 \leqslant \boldsymbol{u}_t^i \leqslant \overline{u}_t^i, 0 \leqslant \boldsymbol{s}_t^i, \quad \forall t\in[T], \tag{5.20c}$$

$$0 \leqslant \boldsymbol{x}_t^i \leqslant \overline{x}_t^i, \quad \forall t\in[T], \tag{5.20d}$$

$$\boldsymbol{u}_t^i \preccurlyeq \mathcal{A}_t \quad \forall t\in[T]. \tag{5.20e}$$

where $i+$ denotes the unique child of i. In particular note that, for a given μ, the subproblem (5.20) can be solved by dynamic programming yielding local one-dimensional value functions V_t^i which take as argument only the local state x_t^i.

The DADP process is summarized in Algorithm 9. As specified, this algorithm produces local value functions V_t^i. Indeed, the local policies are not admissible, as they verify, at convergence, the relaxed coupling constraint, and not the almost sure coupling constraint. However, we can define, for all $t\in[T]$ an approximate coupled value function

$$\widetilde{V}_t : x_t \mapsto \sum_{i=1}^{I} V_t^i(x_t^i)$$

which in turn can be used to define an admissible policy through a coupled forward Bellman operator.

Data: Spatial price process $\lambda^{(0)}$, parameter $r > 0$
Result: Local value function V_t^i
Set $\mu^{(0)} = 0$ **for** $k \in \mathbb{N}$ **do**
  Draw N_{MC} noise scenario ;
  for $i \in [I]$ **do**
    Solve $V^i(\mu^{(0)})$ by DP, returning local value function $V_t^{(k),i}$;
    Simulate N_{MC} local trajectories of $\big(z_t^{(i,j),n}\big)_{t\in[T],n\in[N_{MC}]}$ and $\big(u_t^{i,n}\big)_{t\in[T],n\in[N_{MC}]}$, $\big(s_t^{i,n}\big)_{t\in[T],n\in[N_{MC}]}$ along the noise scenarios.
  for $(i,j)\in E, t\in[T]$ **do**
    Estimate $\Delta_{t,(i,j)}^{(k)} \approx \frac{1}{N_{MC}}\sum_{n=1}^{N_{MC}}\big\{z_t^{(i,j),n} - u_t^{i,n} - s_t^{i,n}\big\}$
  Update spatial price process $\mu^{(k+1)} := \mu^{(k)} + r\Delta^{(k)}$;

Algorithm 9: DADP algorithm for the dam network

5.3.3 Numerical results

We briefly report here some numerical results detailed in [VL7].

On small academic valleys of increasing size, we compare a global Dynamic Programming approach with DADP and SDDP. All these methods produce Bellman functions, whose quality is evaluated with the same forward Bellman operator. The obtained results are given in Table 5.1. The lines "CPU time" corresponds to the time (in minute) needed to compute the Bellman functions (optimization stage only), whereas the lines "value" indicate the cost obtained by Monte Carlo on the coupled model (simulation stage, performed using a 100,000 scenarios sample). The comparisons between the different cost values for the same valley are thus relevant. For both SDDP and DADP, we also give the lower bound corresponding to the Bellman value obtained at the end of the optimization stage.

Valley	**4-Dams**	**6-Dams**	**8-Dams**	**10-Dams**	**12-Dams**
DP CPU time	*1600'*	$\sim 10^8$'	$\sim \infty$	$\sim \infty$	$\sim \infty$
DP value	-3743	N.A.	N.A.	N.A.	N.A.
SDDP CPU time	*6'*	*10'*	*13'*	*50'*	*97'*
SDDP value	-3742	-7027	-11830	-17070	~ -17000
SDDP lower bound	-3754	-7050	-11960	-17260	-19490
DADP CPU time	*7'*	*12'*	*18'*	*24'*	*22'*
DADP value	-3667	-6816	-11570	-16760	~ -17000
DADP lower bound	-3996	-7522	-12450	-17930	-20480
Gap DADP/SDDP	**2.0%**	**3.0%**	**2.2%**	**1.8%**	?

Table 5.1: Results obtained by DP, SDDP and DADP

On larger academic valleys, where the upper bound through simulation is not possible due to the curse of dimensionality, we compare the lower bound obtained by SDDP and DADP in Table 5.2. We notice that DADP seems to be less sensitive than SDDP to the curse of dimensionality.

Valley	**14-Dams**	**18-Dams**	**20-Dams**	**25-Dams**	**30-Dams**
SDDP CPU time	*210'*	*585'*	*970'*	*1560'*	*2750'*
SDDP lower bound	-32024	-46917	-61454	-79440	-100430
DADP CPU time	*40'*	*50'*	*75'*	*140'*	*150'*
DADP lower bound	-32981	-48095	-62802	-80993	-101990

Table 5.2: SDDP and DADP comparison for large academic valleys

Other applications and variations of DADP have since been proposed in [CCDLP20, PDLCC21] for the management of microgrids. In [SAB+20], Seguret *et al.* , in the context of managing numerous thermostatically controlled loads, develop the link with mean-field game, providing improved convergence theory for a specific structure with numerous unit I.

Chapter 6

Games and energy market

In this chapter we consider the impact of risk-aversion on energy market. This work result from a collaboration with Andy Philpott and Henri Gérard, details and proof can be found in [VL8].

We discuss risked competitive partial equilibrium in a setting in which agents are endowed with coherent risk measures. In contrast to social planning models, we show by example that risked equilibria are not unique, even when agents' objective functions are strictly concave. We also show that standard computational methods find only a subset of the equilibria, even with multiple starting points.

6.1 Introduction

Most industrialized regions of the world have over the last thirty years established wholesale electricity markets that take the form of an auction that matches supply and demand. The exact form of these auction mechanisms varies by jurisdiction, but they typically require offers of energy from suppliers at costs they are willing to supply and clear a market by dispatching these offers in order of increasing cost. Day-ahead markets such as those implemented in many North American electricity systems, seek to arrange supply well in advance of its demand so that thermal units can be prepared in time. Since the demand cannot be predicted with absolute certainty, day-ahead markets must be accompanied by a separate balancing market to deal with the variation in load and generator availability in real-time. These are often called *two-settlement* markets. The market mechanisms are designed to be as efficient as possible in the sense that they should aim to maximize the total welfare of producers and consumers.

In response to pressure to reduce CO_2 emissions and increase the penetration of renewables, electricity pool markets are procuring increasing amounts of electricity from intermittent sources such as wind and solar. If probability distributions for intermittent supply are known for these systems then it makes sense to maximize the expected total welfare of producers and consumers in each dispatch. Then many repetitions of this will yield a long-run total benefit that is maximized. Maximizing expected welfare can be modeled as a two-stage stochastic program. Methods for computing prices and single-settlement payment mechanisms for such a *stochastic market clearing* mechanism are described in a number of papers (see [PZP10, WF07, ZPBB16]). When evaluated using the assumed probability distribution on supply, stochastic market clearing can be shown to be more efficient than two-settlement systems.

If agents in these systems are risk-averse then one might also seek to maximize some risk-adjusted social welfare. In this setting, the computation of prices and payments to the agents becomes more complicated. If agents use coherent risk measures then it is possible to define a complete market for risk in a precise sense. If the market is complete then a perfectly competitive partial equilibrium will also maximize risk-adjusted social welfare, i.e. it is efficient. On the other hand, if the market for risk is not complete, then perfectly competitive partial equilibrium can be inefficient. This has been explored in a number of papers (see e.g. [dMdES17, ES11, RS15]).

In this chapter, we study a class of stochastic dispatch and pricing mechanisms under the assumption that agents will attempt to maximize their risk-adjusted welfare at these prices. Agents have coherent risk measures and are assumed to behave as price-takers in the energy and risk markets. We aim at

enlightening some difficulties that arise when risk markets are not complete. We describe a simple instance of a stochastic market that has three different equilibria. Two of these points are stable in the sense of [Sam41] and are attractors of tatônnement algorithms. The third equilibrium is unstable, yet is the solution yielded by the well-known PATH solver in GAMS (See [FM00]). Our example illustrates the delicacy of seeking numerical solutions for equilibria in incomplete markets. Since these are used for justifying decisions, the nonuniqueness of solutions in this setting is undesirable.

The chapter is laid out as follows. In Section 6.2 we present the equilibrium and optimization models we are going to study. In Section 6.3 we give links between equilibrium and optimization problems in the risk-neutral and complete risk-averse cases. Finally, in Section 6.4 we showcase a simple example with multiple equilibria in the incomplete risk-averse case.

6.2 Statement of problem

We consider a probability space $(\Omega, \mathcal{P}(\Omega), \mathbb{P})$ with a finite sample space Ω.

Consider a two time-step single-settlement market for one good. In a single-settlement market, the producer can arrange in advance for a production of x at a marginal cost cx as a first-step decision, and choose the value of a recourse variable $\mathbf{x}_r$ incurring an uncertain marginal cost $\mathbf{c}_r\mathbf{x}_r$. We assume that there are a finite number of scenarios $\omega \in \Omega$ determining the coefficient $\mathbf{c}_r(\omega)$.

The product is purchased in the second step by a consumer with a utility function $\mathbf{V}(\omega)\mathbf{y}(\omega) - \frac{1}{2}\mathbf{r}(\omega)\mathbf{y}^2(\omega)$. The consumer has no first-stage decision, and the amount purchased $\mathbf{y}(\omega)$ depends on the scenario.

6.2.1 Social planner problem

Decisions x, $\mathbf{x}_r(\omega)$ and $\mathbf{y}(\omega)$ can be made to maximize a social objective. We denote by

$$\boldsymbol{W}_p(\omega) = -\frac{1}{2}cx^2 - \frac{1}{2}\mathbf{c}_r(\omega)\mathbf{x}_r(\omega)^2 \ , \tag{6.1a}$$

the welfare of the producer, and by

$$\boldsymbol{W}_c(\omega) = \mathbf{V}(\omega)\mathbf{y}(\omega) - \frac{1}{2}\mathbf{r}(\omega)\mathbf{y}(\omega)^2 \ , \tag{6.1b}$$

the welfare of the consumer where both these expressions ignore the price paid for the good in scenario ω. Then the welfare of the social planner can be defined by $\boldsymbol{W}_{sp} = \boldsymbol{W}_p + \boldsymbol{W}_c$.

Optimization of the social objective requires us to aggregate the uncertain outcomes from the scenarios. This can be done by taking expectations with respect to an underlying probability measure $\mathbb{P}$ or using a more general risk measure.

risk-neutral social planner problem

Endow the set of scenario Ω with a probability $\mathbb{P}$, then a risk-neutral social planner might seek to maximize the expected total social welfare under the constraint that supply equals demand. This problem is denoted by RnSp($\mathbb{P}$) and reads

$$\text{RnSp}(\mathbb{P}) : \max_{x,\mathbf{x}_r,\mathbf{y}} \quad \mathbb{E}_{\mathbb{P}}[\boldsymbol{W}_{sp}] \ , \tag{6.2a}$$

$$\text{s.t.} \quad x + \mathbf{x}_r(\omega) \geqslant \mathbf{y}(\omega) \ , \qquad \forall \omega \in \Omega \ . \tag{6.2b}$$

risk-averse social planner problem

Choosing expectation $\mathbb{E}_{\mathbb{P}}$, assumes a risk-neutral point of view, where two random losses with the same expectation but different variances are deemed equivalent. In practice a number of agents are risk-averse. To model risk aversion we generally use a *risk measure* ρ, which is a functional that associates to

a random welfare its deterministic equivalent, i.e. the deterministic welfare deemed as equivalent to the random loss.

A risk-averse planner solves a maximization problem RaSp(ρ) defined by

$$\text{RaSp}(\rho): \max_{x,\mathbf{x}_r,\mathbf{y}} \quad \rho[\boldsymbol{W}_{sp}]\ , \tag{6.3a}$$

$$\text{s.t.} \quad x + \mathbf{x}_r(\omega) \geqslant \mathbf{y}(\omega)\ , \qquad \forall \omega \in \Omega\ . \tag{6.3b}$$

Recall that a risk measure $\check{\rho}$ is said to be *polyhedral* if its subdifferential at 0 is a polyhedron, in which case Problem RaSp($\check{\rho}$) can be written as follows

$$\text{RaSp}(\check{\rho}): \max_{\theta,x,\mathbf{x}_r,\mathbf{y}} \quad \theta \tag{6.4a}$$

$$\text{s.t.} \quad \theta \leqslant \mathbb{E}_{\mathbb{Q}_k}\big[\boldsymbol{W}_{sp}\big] \qquad \forall k \in [\![1; K]\!]\ , \tag{6.4b}$$

$$x + \mathbf{x}_r(\omega) \geqslant \mathbf{y}(\omega) \qquad \forall \omega \in \Omega\ . \tag{6.4c}$$

In what follows we assume that all risk measures are coherent.

Remark on non-linearity of risk-averse objective function

By linearity of expectation, we have $\mathbb{E}_{\mathbb{P}}[\boldsymbol{W}_{sp}] = \mathbb{E}_{\mathbb{P}}[\boldsymbol{W}_p] + \mathbb{E}_{\mathbb{P}}[\boldsymbol{W}_c]$ hence the criterion of the social planner is natural, which is not the case anymore with risk-aversion. The social planner criterion could be either $\rho[\boldsymbol{W}_{sp}]$ or $\rho[\boldsymbol{W}_p] + \rho[\boldsymbol{W}_c]$. Furthermore, by concavity and positive homogeneity, we have $\rho[\boldsymbol{W}_p + \boldsymbol{W}_c] \geqslant \rho[\boldsymbol{W}_p] + \rho[\boldsymbol{W}_c]$.

6.2.2 Equilibrium problem

We now define a competitive partial equilibrium for our model. This competitive equilibrium can be risk-neutral or risk-averse. Definitions come from general equilibrium theory (See [AD54] or [Uza60]).

risk-neutral equilibrium

Given a probability $\mathbb{P}$ on Ω, a *risk-neutral equilibrium* RnEq($\mathbb{P}$) is a set of prices $\big\{\boldsymbol{\pi}(\omega)\ ,\ \ \omega \in \Omega\big\}$ such that there exists a solution to the system

$$\text{RnEq}(\mathbb{P}):$$

$$\max_{x,\mathbf{x}_r} \quad \mathbb{E}_{\mathbb{P}}\big[\boldsymbol{W}_p + \boldsymbol{\pi}\big(x + \mathbf{x}_r\big)\big]\ , \tag{6.5a}$$

$$\max_{\mathbf{y}} \quad \mathbb{E}_{\mathbb{P}}\big[\boldsymbol{W}_c - \boldsymbol{\pi}\mathbf{y}\big]\ , \tag{6.5b}$$

$$0 \leqslant x + \mathbf{x}_r(\omega) - \mathbf{y}(\omega) \perp \boldsymbol{\pi}(\omega) \geqslant 0\ ,\ \ \forall \omega \in \Omega\ . \tag{6.5c}$$

Here, the producer maximizes its expected profit (6.5a), the consumer maximizes its expected utility (6.5b) and the market clears with (6.5c) (which means that either prices are null or supply equals demand). As the consumer has no first-stage decision, she can optimize each scenario independently and so problem (6.5b) can be replaced by

$$\max_{\mathbf{y}(\omega)} \quad \boldsymbol{W}_c(\omega) - \boldsymbol{\pi}(\omega)\mathbf{y}(\omega)\ ,\ \ \forall \omega \in \Omega\ .$$

risk-averse equilibrium

Given two risk measures ρ_p and ρ_c over Ω, *a risk-averse equilibrium* $\text{RaEq}(\rho_p, \rho_c)$ is a set of prices $\big\{\boldsymbol{\pi}(\omega) : \omega \in \Omega\big\}$ such that there exists a solution to the following system

$$
\begin{aligned}
&\text{RaEq}(\rho_p, \rho_c) : \\
&\max_{x, \mathbf{x}_r} \quad \rho_p\Big[\boldsymbol{W}_p + \boldsymbol{\pi}\big(x + \mathbf{x}_r\big)\Big] \,, && \text{(6.6a)}\\
&\max_{\mathbf{y}} \quad \rho_c\big[\boldsymbol{W}_c - \boldsymbol{\pi}\mathbf{y}\big] \,, && \text{(6.6b)}\\
&0 \leqslant x + \mathbf{x}_r(\omega) - \mathbf{y}(\omega) \perp \boldsymbol{\pi}(\omega) \geqslant 0 \,, \ \ \forall \omega \in \Omega \,. && \text{(6.6c)}
\end{aligned}
$$

Since the coherent risk measure ρ_c of the consumer is monotonic, and noting that she has no first-stage decision, she can optimize scenario per scenario. Thus, she is *insensitive to risk* as any monotonic risk measure will lead to the same action (although not the same welfare). Since ρ_p is also monotonic, we can endow both agents with the same risk measure. In that case, we denote problem (6.6) by $\text{RaEq}(\rho)$.

We now consider polyhedral risk measure $\check{\rho}$, using formulation (6.4), the equilibrium problem (6.6) reads

$$
\begin{aligned}
\text{RaEq}(\check{\rho}) : \max_{\theta, x, \mathbf{x}_r} \quad & \theta && \text{(6.7a)}\\
\text{s.t.} \quad & \theta \leqslant \mathbb{E}_{\mathbb{Q}_k}\big[\boldsymbol{W}_p + \boldsymbol{\pi}(x + \mathbf{x}_r)\big] \,, \ \ \forall k \in [\![1; K]\!] \,, \\
\max_{\mathbf{y}(\omega)} \quad & \boldsymbol{W}_c(\omega) - \boldsymbol{\pi}\mathbf{y}(\omega) \,, \ \ \forall \omega \in \Omega \,, && \text{(6.7b)}\\
& 0 \leqslant x + \mathbf{x}_r(\omega) - \mathbf{y}(\omega) \perp \boldsymbol{\pi}(\omega) \geqslant 0 \,, \ \ \forall \omega \in \Omega \,. && \text{(6.7c)}
\end{aligned}
$$

6.2.3 Trading risk with Arrow-Debreu securities

Until now, we have considered equilibrium problems in an incomplete market. Following the path of [PFW16], we complete the market using Arrow-Debreu securities.

Definition 6.1. *An* Arrow-Debreu security *for node $\omega \in \Omega$ is a contract that charges a price $\mu(\omega)$ in the first-stage, to receive a payment of* 1 *in scenario ω.*

The consumer now has a first-stage decision which is the number of contracts she buys, so the choice of the consumer risk measure ρ_c has no consequences. For convenience, this risk measure ρ_c is chosen to be the same as that of the producer ρ_p and will be denoted by ρ. Unless stated otherwise, from now on we use polyhedral risk measures.

Denote $\mathbf{a}(\omega)$ (resp. $\mathbf{b}(\omega)$) the number of Arrow-Debreu securities bought by the producer (resp. the consumer). We denote by $\boldsymbol{\mu}(\omega)$ the price of the Arrow-Debreu securities associated with scenario ω. In this case the producer pays $\sum_{\omega\in\Omega} \boldsymbol{\mu}(\omega)\mathbf{a}(\omega)$ in the first-stage, in order to receive $\mathbf{a}(\omega)$ in scenario ω. As $\mathbf{a}(\omega) + \mathbf{b}(\omega)$ represents excess demand, requiring that supply is greater than demand consists in requiring $\mathbf{a}(\omega) + \mathbf{b}(\omega) \leqslant 0$. Prices $\{\boldsymbol{\pi}(\omega), \boldsymbol{\mu}(\omega)\}_{\omega\in\Omega}$ form a *risk-trading equilibrium* if there exists a solution to:

$$
\begin{aligned}
&\text{RaEq-AD}(\check{\rho}) : \\
&\max_{\theta, x, \mathbf{x}_r, \mathbf{a}} \quad \theta - \sum_{\omega\in\Omega} \boldsymbol{\mu}(\omega)\mathbf{a}(\omega) && \text{(6.8a)}\\
&\text{s.t.} \quad \theta \leqslant \mathbb{E}_{\mathbb{Q}_k}\big[\boldsymbol{W}_p + \boldsymbol{\pi}(x + \mathbf{x}_r) + \mathbf{a}\big] \,, \ \ \forall k \in [\![1; K]\!] \,, && \text{(6.8b)}\\
&\max_{\phi, \mathbf{y}, \mathbf{b}} \quad \phi - \sum_{\omega\in\Omega} \boldsymbol{\mu}(\omega)\mathbf{b}(\omega) && \text{(6.8c)}\\
&\text{s.t.} \quad \phi \leqslant \mathbb{E}_{\mathbb{Q}_k}\big[\boldsymbol{W}_c - \boldsymbol{\pi}\mathbf{y} + \mathbf{b}\big] \,, \ \ \forall k \in [\![1; K]\!] \,, && \text{(6.8d)}\\
&0 \leqslant x + \mathbf{x}_r(\omega) - \mathbf{y}(\omega) \perp \boldsymbol{\pi}(\omega) \geqslant 0 \,, \ \ \forall \omega \in \Omega \,, && \text{(6.8e)}\\
&0 \leqslant -\mathbf{a}(\omega) - \mathbf{b}(\omega) \perp \boldsymbol{\mu}(\omega) \geqslant 0 \,, \ \ \forall \omega \in \Omega \,. && \text{(6.8f)}
\end{aligned}
$$

6.3 Some equivalences between social planner problems and equilibrium problems

We recall a trivial equivalence between problem RnSp($\mathbb{P}$) and problem RnEq($\mathbb{P}$) before showing an equivalence between problem RaSp($\check{\rho}$) and problem RaEq-AD($\check{\rho}$).

6.3.1 Equivalence in the risk-neutral case

In the risk-neutral case, the social problem is equivalent to the equilibrium problem.

More precisely, let $\mathbb{P}$ be a probability measure over Ω. The elements $x^\sharp$, $\mathbf{x}_r^\sharp$ and $\mathbf{y}^\sharp$ are optimal solutions to RnSp($\mathbb{P}$) if and only if there exist equilibrium prices $\boldsymbol{\pi}^\sharp$ for RnEq($\mathbb{P}$) with associated optimal decisions $x^\sharp$, $\mathbf{x}_r^\sharp$ and $\mathbf{y}^\sharp$.

If both the producer's and the consumer's criterion are strictly concave and if $\mathbb{P}$ charges all ω, then RnSp($\mathbb{P}$) admits a unique solution and RnEq($\mathbb{P}$) admits a unique equilibrium.

6.3.2 Equivalence in the risk-averse case

The following proposition is an extension of Theorem 7 of [RS15], to a model with producers and consumers, in the special case of a finite number of scenarios with polyhedral risk measures. It is obtained by direct manipulation of the KKT conditions of optimality.

Proposition 6.2. *Let $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ be equilibrium prices such that $\left(x^\sharp, \mathbf{x}_r^\sharp, \mathbf{y}^\sharp, \mathbf{a}, \mathbf{b}, \theta, \varphi\right)$ solves RaEq-AD($\check{\rho}$). Then*

(i) $\boldsymbol{\mu}$ is a probability measure, and $x^\sharp, \mathbf{x}_r^\sharp, \mathbf{y}^\sharp$ solves the risk-neutral social planning problem when evaluated using probability $\boldsymbol{\mu}$, RnSp($\boldsymbol{\mu}$).

(ii) $x^\sharp, \mathbf{x}_r^\sharp, \mathbf{y}^\sharp$ solves the risk-averse social planning problem, RaEq-AD($\check{\rho}$) with worst case measure $\boldsymbol{\mu}$.

Remark 6.3. *Note that an equilibrium of RaEq-AD($\check{\rho}$) consists of a price vector $\boldsymbol{\pi}$, giving one price per scenario, and a probability $\boldsymbol{\mu}$ that is seen by both the producer and the consumer as a worst-case probability for the welfare plus trade evaluation.*

Remark 6.4. *In Section 6.4 we give an example of three risked equilibriums without Arrow-Debreu securities, each corresponding to a risk-neutral equilibrium with different measure $\boldsymbol{\mu}(\omega)$. However, if Arrow-Debreu securities are included then two of these equilibria are no longer equilibria in a risk-averse setting. The risk-averse consumer, who without Arrow-Debreu securities had no mechanism to alter his outcomes will trade these securities to improve their risk-adjusted payoff.*

Remark 6.5. *Consider a set of prices $\boldsymbol{\pi}$ that gives a risked equilibrium in which agent i has payoff $\boldsymbol{W}_i(\boldsymbol{\pi})$ and risked payoff $\rho_i\big[\boldsymbol{W}_i(\boldsymbol{\pi})\big]$. Suppose that there exists a probability measure $\mathbb{Q}^*$ such that $\rho_i\big[\boldsymbol{W}_i(\boldsymbol{\pi})\big] = \mathbb{E}_{\mathbb{Q}^*}\Big[\boldsymbol{W}_i(\boldsymbol{\pi})\Big]$. Observe that this does not imply that choosing actions x to maximize $\mathbb{E}[\mathbb{Q}^*]\boldsymbol{W}_i(\boldsymbol{\pi})$ will give $\max_x \rho_i\big[\boldsymbol{W}_i(\boldsymbol{\pi})\big]$. This is because x^* solves*

$$\max_x \rho_i\big[\boldsymbol{W}_i(\boldsymbol{\pi})\big] = \max_x \min_{\mathbb{Q}\in\mathcal{Q}} \mathbb{E}_{\mathbb{Q}}\big[\boldsymbol{W}_i(\boldsymbol{\pi})\big] \ ,$$

and not

$$\max_x \mathbb{E}_{\mathbb{Q}^*}\Big[f_i(x, \boldsymbol{\pi})\Big] \ ,$$

since $\mathbb{Q}^*$ depends on x.

Remark 6.6. *Proposition 6.2 is easily extended to the case where the agents have different risk measures ρ_p and ρ_c with non-disjoint risk set. In this case,*

$$\begin{aligned}\overline{\theta}+\overline{\varphi} &= \min_{\mathbb{Q}_p\in\mathcal{Q}_p}\mathbb{E}_{\mathbb{Q}_p}[\boldsymbol{\pi}\big(x^\sharp+\mathbf{x}_r^\sharp\big)+\boldsymbol{W}_p^\sharp+\overline{\mathbf{a}}] \\ &\quad+\min_{\mathbb{Q}_c\in\mathcal{Q}_c}\mathbb{E}_{\mathbb{Q}_c}[\boldsymbol{W}_c^\sharp-\boldsymbol{\pi}\mathbf{y}^\sharp+\overline{\mathbf{b}}]\,, \\ &\leqslant \min_{\mathbb{Q}\in\mathcal{Q}_p\cap\mathcal{Q}_c}\mathbb{E}_{\mathbb{Q}}[\boldsymbol{W}_c^\sharp+\boldsymbol{W}_p^\sharp]\,, \end{aligned} \tag{6.9}$$

and the social planner uses a risk measure with $\mathcal{Q}=\mathcal{Q}_p\cap\mathcal{Q}c$.

The following proposition (Theorem 11 [PFW16]) stands as a reverse statement for Proposition 6.2.

Proposition 6.7. *Let the elements $x^\sharp$, $\mathbf{x}_r^\sharp$ and $\mathbf{y}_r^\sharp$ be optimal solutions to RaSp($\check{\rho}$), with associated worst case probability measure $\boldsymbol{\mu}$. Then there exists prices $\boldsymbol{\pi}$ such that the couple $(\boldsymbol{\pi},\boldsymbol{\mu})$ forms a risk trading equilibrium for RaEq-AD($\check{\rho}$) with associated optimal solutions $(x^\sharp,\mathbf{x}_r^\sharp,\mathbf{y}^\sharp)$.*

Combining Proposition 6.2 and Proposition 6.7, we are able to state the following result of uniqueness of equilibrium.

Corollary 6.8. *If both the producer's and consumer's criterion are strictly concave, and if each of the extreme points $\mathbb{Q}_k$ charges all ω, then RaSp($\check{\rho}$) admits a unique solution $(x^\sharp,\mathbf{x}_r^\sharp,\mathbf{y}^\sharp)$. Furthermore, RaEq-AD($\check{\rho}$) admits unique optimal decisions $(x^\sharp,\mathbf{x}_r^\sharp,\mathbf{y}^\sharp)$. If, in addition, solving RaSp($\check{\rho}$) admit a unique worst case probability measure $\boldsymbol{\mu}$, then equilibrium prices $(\boldsymbol{\pi},\boldsymbol{\mu})$ are unique.*

We have shown a first equivalence between RnSp($\mathbb{P}$) and RnEq($\mathbb{P}$) and a second one between RaSp($\check{\rho}$) and RaEq-AD($\check{\rho}$). These equivalences lead to uniqueness of equilibrium if there is uniqueness of the solution of the social planner. A natural question arises: if RaSp($\check{\rho}$) has a unique solution, is there a unique equilibrium for RaEq($\check{\rho}$)? The next section provides a simple counterexample.

6.4 Multiple risk-averse equilibrium

In this section, we present a toy problem where RaSp($\check{\rho}$) has a unique optimum but there are three different equilibria for RaEq($\check{\rho}$). This equilibrium has been found both with classical methods (PATH solver ([FDJM09, FM00]) and a tâtonnement algorithm), then derived analytically. An interesting point is that the equilibrium found by PATH is unstable.

Let $\Omega=\{1,2\}$ and $\mathcal{Q}=\mathrm{conv}\big\{(\frac{1}{4},\frac{3}{4}),(\frac{3}{4},\frac{1}{4})\big\}$. For simplicity of notation index by $i\in\{1,2\}$ the realization of each random variable. We choose the following parameters: $V_1=4$, $V_2=\frac{48}{5}$, $c=\frac{23}{2}$, $c_1=1$, $c_2=\frac{7}{2}$, $r_1=2$, $r_2=10$.

First, we run the PATH solver from different starting points and always found the equilibrium $\boldsymbol{\pi}=(\pi_1,\pi_2)=(1.23578;2.10953)$ leading to risked adjusted welfare $(2.134;0.821)$ for producer and consumer respectively (In blue in Fig. 6.2).

Then, compute the equilibrium using a tâtonnement algorithm (See [Uza60])[1]

Starting from $(1.25;2.06)$, respectively and $(1.22;2.18)$, we and found two new equilibria:

$$\boldsymbol{\pi}=(1.2256;2.0698)\text{ and }\boldsymbol{\pi}=(1.2478;2.1564)\,,$$

leading to risked-adjusted welfare for producer and consumer respectively $(2.152;0.798)$ and $(2.113;0.845)$. Notice that neither equilibrium dominates the other.

Finally, we can analytically study this toy problem.

[1]For given prices $(\pi_1^{(k)},\pi_2^{(k)})$, find $(x^{(k)},x_1^{(k)},x_2^{(k)})$ that maximizes $\rho\big[\boldsymbol{W}_p(\pi_1^{(k)},\pi_2^{(k)})\big]$ and $(y_1^{(k)},y_2^{(k)})$ that maximizes $\rho\big[\boldsymbol{W}_c(\pi_1^{(k)},\pi_2^{(k)})\big]$. Then update the prices with $\pi_i^{(k+1)}=\pi_i^{(k)}-\tau\max(0,y_i-(x+x_i))$.

	condition	$x^\sharp$	$x_i^\sharp$	$y_i^\sharp$
case a)	$x_c \leqslant \frac{\mathbb{E}\left[\overline{p}\right]\boldsymbol{\pi}}{c}$	$\frac{\mathbb{E}\left[\overline{p}\right]\boldsymbol{\pi}}{c}$	$\frac{\pi_i}{c_i}$	$\frac{V_i-\pi_i}{r_i}$
case b)	$\frac{\mathbb{E}\left[\overline{p}\right]\boldsymbol{\pi}}{c} \leqslant x_c \leqslant \frac{\mathbb{E}\left[\underline{p}\right]\boldsymbol{\pi}}{c}$	x_c	$\frac{\pi_i}{c_i}$	$\frac{V_i-\pi_i}{r_i}$
case c)	$\frac{\mathbb{E}\left[\underline{p}\right]\boldsymbol{\pi}}{c} \leqslant x_c$	$\frac{\mathbb{E}\left[\underline{p}\right]\boldsymbol{\pi}}{c}$	$\frac{\pi_i}{c_i}$	$\frac{V_i-\pi_i}{r_i}$

Table 6.1: Optimal control for producer and consumer problems

Consider two probabilities $(\underline{p}, 1-\underline{p})$ and $(\overline{p}, 1-\overline{p})$ Given prices $0 < \pi_1 < \pi_2$, we solve the producer (resp. consumer) optimization problem. Optimal decisions are derived in **??** and summed up in Table 6.1 where x_c is given by

$$x_c(\boldsymbol{\pi}) = \frac{1}{2(\pi_1 - \pi_2)}\left[\frac{\pi_2^2}{c_2} - \frac{\pi_1^2}{c_1}\right] .$$

We see that there are three regimes, depending only on the prices (π_1, π_2), of optimal first-stage solutions. Case a) (resp. case c)), corresponds to a set of prices such that $\mathbb{E}_{\overline{p}}[\boldsymbol{W}_p] < \mathbb{E}_{\underline{p}}[\boldsymbol{W}_p]$ (resp. $\mathbb{E}_{\overline{p}}[\boldsymbol{W}_p] > \mathbb{E}_{\underline{p}}[\boldsymbol{W}_p]$), and the optimal decision corresponds to an optimal risk-neutral decision with respect to one of the two extreme points of $\mathcal{Q}$. On the other hand, case b) corresponds to a set of prices such that the expected welfare is equivalent for all probability in $\mathcal{Q}$, i.e. $\mathbb{E}_{\overline{p}}[\boldsymbol{W}_p] = \mathbb{E}_{\underline{p}}[\boldsymbol{W}_p]$. In Figure 6.1, the red area corresponds to case a), the blue to case b) and the red to case c), separated by black lines of equations $\frac{\mathbb{E}[\overline{p}]\boldsymbol{\pi}}{c} = x_c(\boldsymbol{\pi})$ and $\frac{\mathbb{E}[\underline{p}]\boldsymbol{\pi}}{c} = x_c(\boldsymbol{\pi})$ respectively.

We are now looking for prices (π_1, π_2) such that the complementarity constraints are satisfied. For strictly positive prices, these constraints can be summed up as

$$z_i(\boldsymbol{\pi}) = x^\sharp(\boldsymbol{\pi}) + x_i^\sharp(\boldsymbol{\pi}) - y_i^\sharp(\boldsymbol{\pi}) = 0 , \qquad i \in \{1, 2\}.$$

Accordingly, we define excess supply functions z_i^l for case $l \in \{a, b, c\}$, and $i \in \{1, 2\}$. The red, blue and green lines correspond to manifolds of null excess supply function for scenario i, that is of prices such that $z_i^l(\pi_1, \pi_2) = 0$. When the lines cross we have $z_l^1 = z_l^2 = 0$, and thus we have candidate equilibrium. If the lines cross in the area of the same color we have an equilibrium. This is the case with the parameters chosen, and equilibrium can be derived in exact arithmetic.

We end with a few remarks derived from this example.

Remark 6.9. *The PATH solver finds the blue equilibrium, tatônnement algorithm finds the green and the red equilibrium as illustrated by Figure 6.2. Interestingly it can be shown that the blue equilibrium is unstable in the sense that the dynamical system driven by $\pi' = z(\pi)$ is not locally stable (see [Sam41]) around the blue equilibrium.*

Remark 6.10. *No equilibrium dominates another: if going from one equilibrium to another increases the (risk-adjusted) welfare of one agent, then it decreases the (risk-adjusted) welfare of the other.*

Remark 6.11. *Using the analytical results we check that there exists a set of non-zero Lebesgue measure of parameters $V_1, V_2, c, c_1, c_2, r_1$, and r_2 (albeit small), that have three distinct equilibria with the same properties.*

Remark 6.12. *We can show that the blue equilibrium is a convex combination of red and green equilibrium, illustrated in Figure 6.1 by the dashed blue line.*

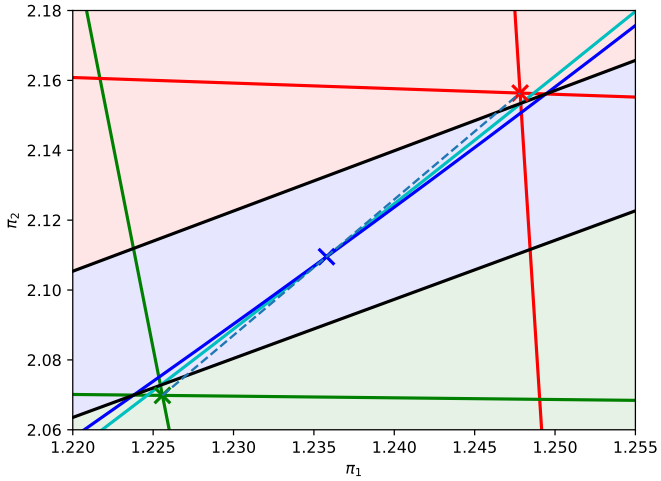
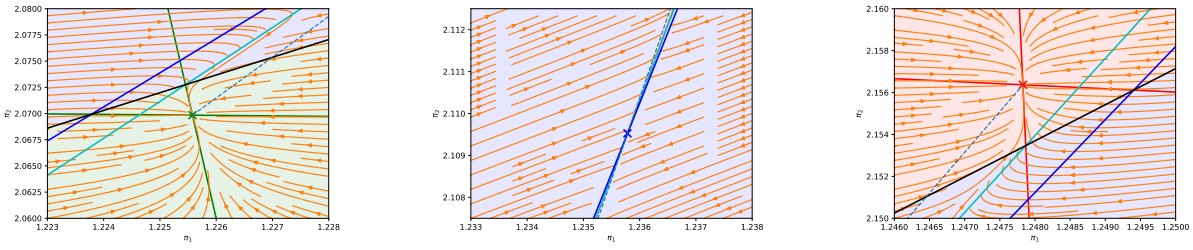


Figure 6.1: Null excess function per scenario manifold for $V_1 = 4$, $V_2 = \frac{48}{5}$, $c = \frac{23}{2}$, $c_1 = 1$, $c_2 = \frac{7}{2}$, $r_1 = 2$, $r_2 = 10$.



(a) around green equilibrium. (b) around blue equilibrium. (c) around red equilibrium.

Figure 6.2: Representation of vector field $\pi' = z(\pi)$

Chapter 7

Applications in supply chain management

In this chapter, we address a production problem modeled as a stochastic multi-item lot-sizing problem with bounded numbers of setups per period and without setup cost, cast as an MSP. While this formulation seems to be rather non-standard in the lot-sizing landscape, it is motivated by concrete missions of an industrial partner company. This work was carried as part of the CIFRE Ph.D. Thesis of Étienne Gaillard de Saint-Germain, co-supervised with F. Meunier in partnership with Argon Consulting. It was presented in [VL6], and more complete results are to be found in [dSG18].

7.1 Problem motivation and formulation

Fixing the production level for the forthcoming week is a basic decision to be taken when managing a production line. Usually, demand has to be satisfied at due dates but the limited capacity of the line prevents last-minute production. On the other hand, too early productions may lead to unnecessary high inventory costs. The challenge of this kind of problem, known as *lot-sizing problems*, consists in finding a trade-off between demand satisfaction and holding costs. This is a well-studied topic, with many variations (deterministic/stochastic, single/multi-item, etc.). When several references can be produced on the same line (*multi-item*), the capacity is often all the more reduced as the number of distinct references produced over the current week is high. Indeed, changing a reference in production stops the line for a moment. This additional capacity reduction is usually modeled by setup costs contributing to the total cost. We consider a stochastic multi-item lot-sizing problem where the capacity reduction due to reference changes is not modeled by setup costs but instead by an explicit upper bound on the total number of references that can be produced over a week. This represents a stochastic variation of the Capacited Lot-Sizing Problem.

The assembly line produces a set $\mathcal{R}$ of references over T weeks. The number of distinct references produced over a week cannot exceed N. There is also an upper bound on the total week production (summed over all references). We normalize all quantities so that this upper bound is equal to 1. The production of each reference r must satisfy a random demand $\boldsymbol{d}_t^r$ over week t. Inventory can be stored but incurs a unit holding cost $h^r > 0$ per week. Note that there is no setup cost. For each reference r, there is an initial inventory $s_0^r \in \mathbb{R}_+$.

Finally, due to uncertainty or limited capacity, the demand is not always met. We first consider an optimization problem under a service level constraint and then consider backorder costs. The later formulation can be seen as an approximation of the first, and we offer a way to determine backorder costs.

7.1.1 Model with service level constraint

For each period t and each reference r, we introduce the decision variable $\widetilde{\boldsymbol{d}}_t^r$ which is the part of demand $\boldsymbol{d}_t^r$ satisfied at the end of period t. We decide to model the service level constraint for all items by

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}} w^r \frac{\sum_{t=1}^T \widetilde{\boldsymbol{d}}_t^r}{\sum_{t=1}^T \boldsymbol{d}_t^r}\right] \geq \beta \quad \text{where} \quad w^r = \frac{h^r}{\sum_{j\in\mathcal{R}} h^j}. \tag{7.1}$$

Then, we can write the risk-neutral MSP (1.6) corresponding to our problem at time t

$$\begin{aligned}
\text{Min} \quad & \mathbb{E}\left[\sum_{t'=t}^{T}\sum_{r\in\mathcal{R}} h^r \boldsymbol{s}_{t'}^r\right] && && \text{(7.2a)}\\
\text{s.t.} \quad & \boldsymbol{s}_{t'}^r = \boldsymbol{s}_{t'-1}^r + \boldsymbol{q}_{t'}^r - \widetilde{\boldsymbol{d}}_{t'}^r && \forall t' \in [\![t,T]\!],\ \forall r\in\mathcal{R}, && \text{(7.2b)}\\
& \sum_{r\in\mathcal{R}} \boldsymbol{q}_{t'}^r \leqslant 1 && \forall t' \in [\![t,T]\!], && \text{(7.2c)}\\
& \boldsymbol{q}_{t'}^r \leqslant \boldsymbol{x}_{t'}^r && \forall t' \in [\![t,T]\!],\ \forall r\in\mathcal{R}, && \text{(7.2d)}\\
& \sum_{r\in\mathcal{R}} \boldsymbol{x}_{t'}^r \leqslant N && \forall t' \in [\![t,T]\!], && \text{(7.2e)}\\
& \mathbb{E}\left[\sum_{r\in\mathcal{R}} w^r \frac{\sum_{t=1}^T \widetilde{\boldsymbol{d}}_t^r}{\sum_{t=1}^T \boldsymbol{d}_t^r}\right] \geqslant \beta && && \text{(7.2f)}\\
& \widetilde{\boldsymbol{d}}_{t'}^r \leqslant \boldsymbol{d}_{t'}^r && \forall t' \in [\![t,T]\!],\ \forall r\in\mathcal{R}, && \text{(7.2g)}\\
& \boldsymbol{x}_{t'}^r \in \{0,1\} && \forall t' \in [\![t,T]\!],\ \forall r\in\mathcal{R}, && \text{(7.2h)}\\
& \boldsymbol{q}_{t'}^r,\ \boldsymbol{s}_{t'}^r,\ \widetilde{\boldsymbol{d}}_{t'}^r \geqslant 0 && \forall t' \in [\![t,T]\!],\ \forall r\in\mathcal{R}, && \text{(7.2i)}\\
& \boldsymbol{q}_{t'}^r \preccurlyeq \sigma\left(\left(\boldsymbol{d}_1^r,\ldots,\boldsymbol{d}_{t'-1}^r\right)_{r\in\mathcal{R}}\right) && \forall t' \in [\![t,T]\!],\ \forall r\in\mathcal{R}. && \text{(7.2j)}
\end{aligned}$$

Objective (7.2a) minimizes the future expected holding costs. Constraint (7.2b) is the inventory dynamic. Capacity of the assembly line is ensured by constraint (7.2c). Constraint (7.2d) is both a "big-M" constraint and a capacity of the production of a single item. Constraint (7.2e) limits the number of setups at each period. Constraint (7.2f) ensures the service level. Constraint (7.2g) means that we cannot satisfy more than the demand. Last constraint (7.2j) is the non-anticipativity constraint. Every constraint of the problem, except the service level constraint (7.2f), holds almost surely.

Note that, even with the stagewise independent node Assumption (SWI), this problem cannot be tackled by dynamic programming due to Constraint (7.2g). Even without this constraint, vanilla Dynamic Programming approaches (see Algorithm 1) fail due to high dimensionality, and SDDP (see Chapter 3) do not apply due to binary variables.

This formulation matches our industrial partner's objective, but is not always feasible, because of the service level constraint (7.2f). For this reason, as well as to have a formulation without time-coupling constraints, we consider an alternative model with backorder costs. This new model can be seen as a proxy of the previous one —in which case we suggest a way of setting the backorder costs— or be of interest in itself.

7.1.2 Model with backorder costs

Instead of including a service level constraint, we decide to allow backorder and penalize it. We introduce new decision variables. When a demand for reference r is not satisfied by the production of the current period or by inventory, it can be satisfied later but incurs a unit backorder cost γ^r per period for some coefficient $\gamma^r > 0$ and the backorder of reference r at the end of the period t is denoted by $\boldsymbol{b}_t^r$.

The problem at time t can be written as follow.

$$\begin{aligned}
\text{Min} \quad & \mathbb{E}\left[\sum_{t'=t}^{T}\sum_{r\in\mathcal{R}}(h^r \boldsymbol{s}^r_{t'} + \gamma^r \boldsymbol{b}^r_{t'})\right] && && \text{(7.3a)}\\
\text{s.t.} \quad & \widetilde{\boldsymbol{s}}^r_{t'} = \widetilde{\boldsymbol{s}}^r_{t'-1} + \boldsymbol{q}^r_{t'} - \boldsymbol{d}^r_{t'} && \forall t' \in [\![t,T]\!],\ \forall r\in\mathcal{R}, && \text{(7.3b)}\\
& \sum_{r\in\mathcal{R}} \boldsymbol{q}^r_{t'} \leqslant 1 && \forall t' \in [\![t,T]\!], && \text{(7.3c)}\\
& \boldsymbol{q}^r_{t'} \leqslant \boldsymbol{x}^r_{t'} && \forall t' \in [\![t,T]\!],\ \forall r\in\mathcal{R}, && \text{(7.3d)}\\
& \sum_{r\in\mathcal{R}} \boldsymbol{x}^r_{t'} \leqslant N && \forall t' \in [\![t,T]\!], && \text{(7.3e)}\\
& \widetilde{\boldsymbol{s}}^r_{t'} = \boldsymbol{s}^r_{t'} - \boldsymbol{b}^r_{t'} && \forall t' \in [\![t,T]\!],\ \forall r\in\mathcal{R}, && \text{(7.3f)}\\
& \boldsymbol{x}^r_{t'} \in \{0,1\} && \forall t' \in [\![t,T]\!],\ \forall r\in\mathcal{R}, && \text{(7.3g)}\\
& \boldsymbol{q}^r_{t'},\ \boldsymbol{s}^r_{t'},\ \boldsymbol{b}^r_{t'} \geqslant 0 && \forall t' \in [\![t,T]\!],\ \forall r\in\mathcal{R}, && \text{(7.3h)}\\
& \boldsymbol{q}^r_{t'} \preccurlyeq \sigma\left(\left(\boldsymbol{d}^r_1,\dots,\boldsymbol{d}^r_{t'-1}\right)_{r\in\mathcal{R}}\right) && \forall t' \in [\![t,T]\!],\ \forall r\in\mathcal{R}. && \text{(7.3i)}
\end{aligned}$$

An interesting feature of this model is that there always exists a feasible solution, which makes it more amenable to real-world applications. However, it cannot guarantee a specified service level.

When backorder costs are not given by the clients, we propose a way to "price" backorder coefficients γ^r for each reference r before the first period, with the idea to heuristically drive the model to choose solutions satisfying service level constraint (7.2f). We set

$$\gamma^r \coloneqq \frac{\mathbb{P}\left[\boldsymbol{d}^r \leqslant q^r(\beta)\right]}{\mathbb{P}\left[\boldsymbol{d}^r > q^r(\beta)\right]} h^r \tag{7.4}$$

with

$$q^r(\beta) \coloneqq \inf\left\{ q\in\mathbb{R}_+ \;\middle|\; \mathbb{E}\left[\frac{\min(\boldsymbol{d}^r, q)}{\boldsymbol{d}^r}\right] \geqslant \beta \right\} \tag{7.5}$$

where $\boldsymbol{d}^r = \sum_{t=1}^{T} \boldsymbol{d}^r_t$ is the demand of reference r aggregated over time. Since $\boldsymbol{d}^r$ is non-negative, $q^r(\beta)$ is well-defined (we set $\frac{0}{0} = \beta$, so that items with no demand would not impact the constraint). Computing an approximate value of $q^r(\beta)$ at an arbitrary precision can easily be performed by binary search.

To justify this choice, consider the second problem (7.3) with only one item and for a horizon of one period. Assuming no initial inventory, it takes then the form of the famous *newsvendor problem*

$$\min_{q\geqslant 0} \quad \mathbb{E}\left[h^r(q-\boldsymbol{d}^r)^+ + \gamma^r(\boldsymbol{d}^r - q)^+\right], \tag{7.6}$$

where γ^r is a unit backorder cost specific to reference r.

Remark 7.1. *If instead of controlling the* fill rate service level*, we want to control the* cycle service level*, defined as the probability of satisfying the whole demand, then we can choose*

$$\gamma^r = \frac{\beta}{1-\beta} h^r. \tag{7.7}$$

Indeed, in this case, the optimal solution q^{r*} of (7.6) *satisfies $\mathbb{P}(q^{r*} \geqslant \boldsymbol{d}^r) = \beta$. Interestingly, Eq.* (7.7) *does not depend on the distribution of the demand, which contrasts with Eq.* (7.4)*.*

7.2 Solution method

We now propose a method to solve the CLSP problem with backorder (7.3). As even its deterministic version is challenging (see Remark 7.2 below.), we turn to heuristic methods. The generic idea is to solve

a (simplified) problem at time $t = 1$, decide the production variables for the first week, then reveal the actual demand realization and evaluate the inventory at the end of week 1. We can then solve the problem starting at $t = 2$ to determine the decision to be taken for week 2, and so on.

Remark 7.2 (Deterministic problem challenges)**.** *Note that the deterministic problem is already challenging. Indeed, it is NP-hard in the strong sense for any fixed $N \geqslant 3$, since there is a straightforward reduction from 3-PARTITION, for $N \geqslant 3$. The complexity status of the deterministic version when $N = 1$ or $N = 2$ seems to be a challenging open question.*

It is also worth noting that the optimal value of the continuous relaxation is independent of N. In an attempt of improving the quality of the continuous relaxation, one may consider the extended formulation with the binary variables y_t^p for each $p \in \binom{\mathcal{R}}{N}$ and each $t \in [T]$ in place of the x_t^r's (indicating the references produced on period t). Alternatively, we can consider the extended formulation with the binary variables z_τ^r for each $\tau \subseteq [T]$ (indicating the periods of production of the reference r). In both cases, we can show that this does not improve the quality of the linear relaxation.

Remark 7.3 (Dynamic Programming)**.** *Problem* (7.3) *is an MSP. As such, if the demand process is assumed to be stagewise independent (or with limited memory), they satisfy some Bellman's equation, and could be tackled by Dynamic Programming approaches. Unfortunately, as seen in Remark 7.2, the binary variables here are key, and the relaxed version might not be very informative. Thus, classical SDDP approaches are not relevant, and their integer counterpart often slow in practice. Nevertheless, some recent work [QGKS22] has been able to use SDDiP, with dedicated improvements, on a related problem.*

The heuristic presented here is a repeated two-stage approach (or stochastic model predictive control). More precisely, for each stage t, we solve a two-stage approximation of Problem (7.3), denoted (2SA), where the non-anticipativity constraint (7.3i) is relaxed into

$$\begin{cases} \sigma\left(\boldsymbol{q}_t^r\right) \subset \sigma\left(\varnothing\right) & r \in \mathcal{R} \\ \sigma\left(\boldsymbol{q}_{t'}^r\right) \subset \sigma\left(\left(\boldsymbol{d}_t^{r'}, \ldots, \boldsymbol{d}_T^{r'}\right)_{r' \in \mathcal{R}}\right) & t' \geqslant t+1,\ r \in \mathcal{R}. \end{cases}$$

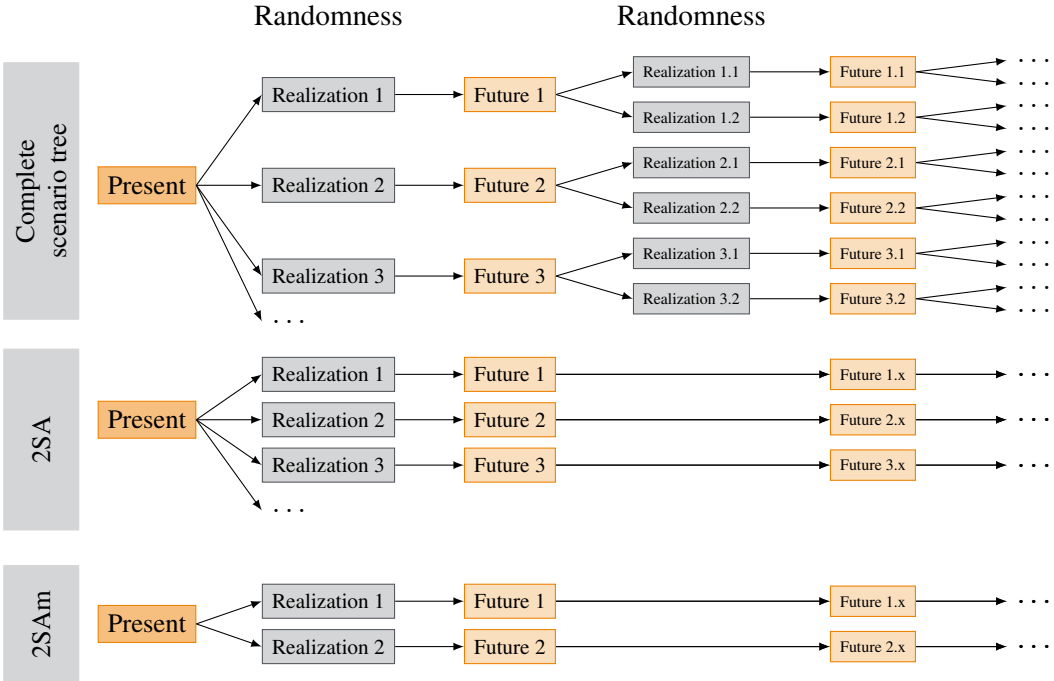


Figure 7.1: Scheme of the scenario approximation

The (2SA) relaxation is then approximated by a classical sample average approximation, illustrated in Fig. 7.1. More precisely we draw a set Ξ of m sampled Ξ, each of these scenarios representing

a possible realization of $(\boldsymbol{d}_t^r, \boldsymbol{d}_{t+1}^r, \dots, \boldsymbol{d}_T^r)$ for each r, and solve, with any standard MIP solver, the following mixed integer program (2SA-m).

$$\min \quad \frac{1}{m} \sum_{\omega \in \Xi} \sum_{t'=t}^{T} \sum_{i \in \mathcal{R}} \left(h^r s_{t',\omega}^r + \gamma^r b_{t',\omega}^r \right) \tag{7.8a}$$

$$\text{s.t.} \quad \widetilde{s}_{t',\omega}^r = \widetilde{s}_{t'-1,\omega}^r + q_{t',\omega}^r - d_{t',\omega}^r \qquad \forall \omega \in \Xi,\ \forall t' \in [\![t, T]\!],\ \forall r \in \mathcal{R}, \tag{7.8b}$$

$$\sum_{i \in \mathcal{R}} q_{t',\omega}^r \leqslant 1 \qquad \forall \omega \in \Xi,\ \forall t' \in [\![t, T]\!], \tag{7.8c}$$

$$q_{t',\omega}^r \leqslant x_{t',\omega}^r \qquad \forall \omega \in \Xi,\ \forall t' \in [\![t, T]\!],\ \forall r \in \mathcal{R}, \tag{7.8d}$$

$$\sum_{i \in \mathcal{R}} x_{t',\omega}^r \leqslant N \qquad \forall \omega \in \Xi,\ \forall t' \in [\![t, T]\!], \tag{7.8e}$$

$$\widetilde{s}_{t',\omega}^r = s_{t',\omega}^r - b_{t',\omega}^r \qquad \forall \omega \in \Xi,\ \forall t' \in [\![t, T]\!],\ \forall r \in \mathcal{R}, \tag{7.8f}$$

$$x_{t,\omega}^r = x_t^r \qquad \forall \omega \in \Xi,\ \forall r \in \mathcal{R}, \tag{7.8g}$$

$$q_{t,\omega}^r = q_t^r \qquad \forall \omega \in \Xi,\ \forall r \in \mathcal{R}, \tag{7.8h}$$

$$x_t^r,\ x_{t',\omega}^r \in \{0, 1\} \qquad \forall \omega \in \Xi,\ \forall t' \in [\![t, T]\!],\ \forall r \in \mathcal{R}, \tag{7.8i}$$

$$q_t^r,\ q_{t',\omega}^r,\ s_{t',\omega}^r,\ b_{t',\omega}^r \geqslant 0 \qquad \forall \omega \in \Xi,\ \forall t' \in [\![t, T]\!],\ \forall r \in \mathcal{R}, \tag{7.8j}$$

The validity of this method for solving (2SA) is supported by the standard theory of SAA (see [SDR14]), declined in the following proposition.

Proposition 7.4. *The following three properties hold when m goes to infinity:*

(i) *The value of* (2SA-m) *converges almost surely to the optimal value of* (2SA).

(ii) *For every m, we consider the values $(\hat{q}_{t,m}^r, \hat{x}_{t,m}^r)_{r \in \mathcal{R}}$ of the decision variables for week t of an optimal solution of* (2SA-m). *Any limit point of these values is an optimal solution of* (2SA).

(iii) *Let $\varepsilon > \delta > 0$. Assume that the random demand $(\boldsymbol{d}_{t'}^r)_{t' \geqslant t, r \in \mathcal{R}}$ is such that*

$$\exists C, K, \quad \forall u \in \mathbb{R}, \qquad \mathbb{E}[e^{u\|\boldsymbol{d}\|}] \leqslant C e^{u^2 K}. \tag{7.9}$$

Denote by $\mathcal{Q}_m^\delta$ (resp. $\mathcal{Q}^\varepsilon$) the set of all possible values of $(\hat{q}_{t,m}^r)_{r \in \mathcal{R}}$ in a δ-optimal solution of (2SA-m) *(resp. in an ε-optimal solution of* (2SA)*). Then for every $\alpha \in (0, 1)$, we have $\mathbb{P}(\mathcal{Q}_m^\delta \subseteq \mathcal{Q}^\varepsilon) > 1 - \alpha$ for m large enough.*

If the random demand $\boldsymbol{d}$ is bounded or Gaussian then it satisfies (7.9). For our results we used an expanded Dirichlet distribution which satisfies (7.9).

7.3 Numerical experiments

The instances used are realistic and have been provided by a client of the partner. We consider two lines, one experiencing overcapacity (line L2), and the other experiencing under capacity. More complete results can be found in [dSG18, Chap. 7].

Remark 7.5 (Expanded Dirichlet distribution)**.** *The expanded Dirichlet distribution used in [dSG18, chapter 7] to generate random demand is well suited to our purpose as it allows us to define the expected demand for each product, and an additional volatility parameter that scales the covariance matrix. Further, the sum over all products is constant, thus underlining the flexibility of each approach to adapt to market distribution instead of being driven by the global capacity of the line. Moreover, as conditional expanded Dirichlet distribution is an expanded Dirichlet distribution, and as it is easy to sample an expanded Dirichlet distribution, we can easily generate scenarios for the problem at time t knowing the past demand realizations.*

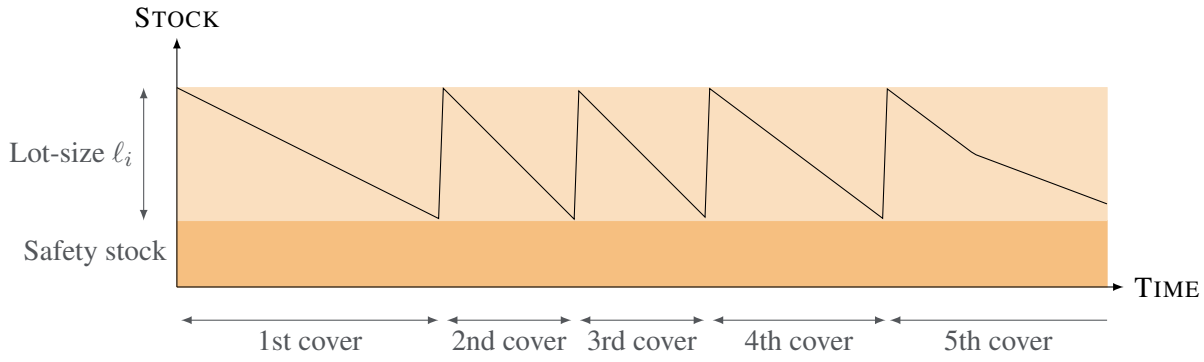
7.3.1 Benchmark heuristics

We present here three other heuristics used to benchmark our approach.

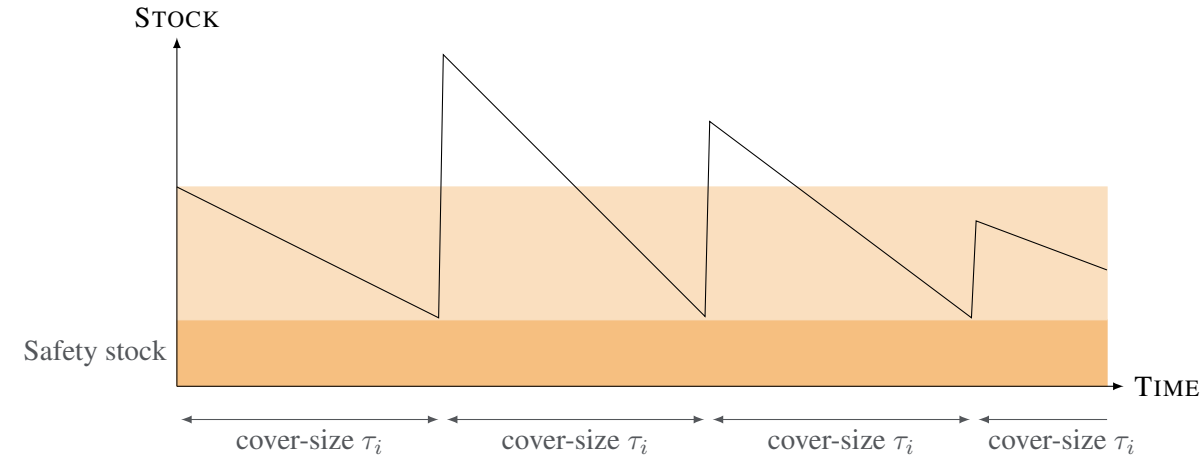
The first heuristic is a Model Predictive control approach, where we solve at time t a deterministic version of (S), where the random demand is replaced by its expectation.

The second one, the *lot-size heuristic*, consists in determining before the first week once and for all a value ℓ_r^* for each reference $r \in \mathcal{R}$. At time t, if the inventory of reference r is below a precomputed safety level, the quantity q_t^r is chosen so that the inventory of reference r exceeds the safety level of exactly ℓ_r^*. In case of capacity issues, the production is postponed and thus backorder costs appear. In addition, if some capacity issues are easily anticipated, the production of a reference r can be activated even if the inventory is not below the safety level.

The third one, the *cover-size heuristic* is almost the same, but instead of precomputing a fixed quantity for each reference, a duration τ_r^* is fixed before the first week. When the inventory of reference r is below the safety level, the quantity q_t^r is computed so that the inventory of reference r exceeds the safety level of the expected demand for the next τ_r^* weeks. These heuristics are illustrated in Fig. 7.2.



(a) lot-size



(b) Cover-size

Figure 7.2: Computation of produced quantities using lot-size and cover-size heuristics

The values ℓ_r^* and τ_r^* are determined as follows. $(\tau_r^*)_{r\in\mathcal{R}}$ is actually chosen to be the optimal solution of the following convex program, which somehow considers the problem at a "macroscopic" level. (Sim-

ilar convex programs in the same context have been considered in the literature; see [**?**] for example.)

$$\begin{aligned} \min \quad & \sum_{r\in\mathcal{R}} h^r \overline{d}^r \tau_r \\ \text{s.t.} \quad & \sum_{r\in\mathcal{R}} \frac{1}{\tau_r} \leqslant N \\ & \tau_r > 0 \qquad\qquad r \in \mathcal{R}, \end{aligned}$$

where $\overline{d}^r = \mathbb{E}\big[\sum_{t=1}^{T} \boldsymbol{d}_t^r\big]$.

The parameter ℓ_r^* of the lot-size heuristic is then set to $\overline{d}^r \tau_r^*$.

7.3.2 Numerical results

We present here numerical results on industrial data. The demand is generated through a stochastic process with a "volatility" parameter $v \in \{20, 50\}$, the smaller value representing more predictable demand. We consider three backorder costs γ. Details are given in [§5.1 VL6] or [dSG18].

The results are provided in Table 7.1. All quantities are given with a confidence interval of 95%. The Column LB provides the lower bound obtained by the optimal value at time $t = 1$ of program (2SA-m) (with $m = 1000$ and a time limit of 24 hours for the solver). The column 2SA-m is the estimated cost of the method proposed in Section 7.2. (We remind the reader that we propose $m = 20$ in this case.) The next three columns provide the results for the three heuristics described in Section 7.3.1.

Instances	LB	2SA-m	Det.	Cover-size	Lot-size
L2_v20_13	0.53	0.89 ± 0.03	1.17 ± 0.10	6.95 ± 0.17	7.79 ± 0.14
L2_v20_81	0.94	2.29 ± 0.06	2.36 ± 0.07	8.12 ± 0.19	9.65 ± 0.14
L2_v20_203	1.00	3.05 ± 0.07	3.25 ± 0.08	9.35 ± 0.29	10.99 ± 0.19
L2_v50_48	0.97	2.73 ± 0.11	3.06 ± 0.21	8.03 ± 0.26	8.37 ± 0.21
L2_v50_154	1.36	4.54 ± 0.20	5.06 ± 0.33	10.83 ± 0.53	11.20 ± 0.38
L2_v50_341	1.51	5.91 ± 0.25	7.90 ± 0.66	15.17 ± 1.21	14.65 ± 0.77
L6_v20_3	0.54	0.61 ± 0.01	0.70 ± 0.02	1.71 ± 0.08	1.74 ± 0.08
L6_v20_19	1.41	1.81 ± 0.06	1.86 ± 0.06	3.51 ± 0.12	3.20 ± 0.08
L6_v20_55	2.67	3.57 ± 0.24	3.71 ± 0.30	7.49 ± 0.39	6.24 ± 0.34
L6_v50_11	1.33	2.00 ± 0.11	2.14 ± 0.12	3.42 ± 0.15	3.03 ± 0.13
L6_v50_42	2.99	4.45 ± 0.53	4.48 ± 0.51	7.99 ± 0.62	6.57 ± 0.60
L6_v50_98	6.13	8.29 ± 1.23	7.94 ± 1.04	16.34 ± 1.61	12.96 ± 1.45

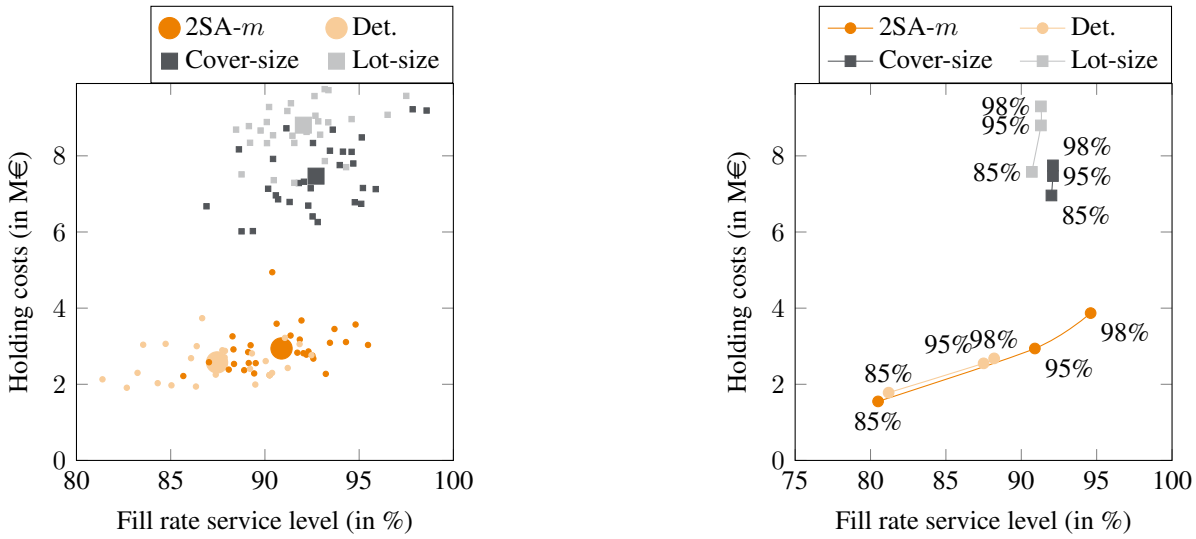
Table 7.1: Results - Inventory costs (in M€)

Our method clearly outperforms lot-size and cover-size heuristics and is better than the deterministic approximation for all but one instance. By running our method instead of a usual heuristic at the beginning of each week, the inventory costs can be reduced often by more than 50%. For the instance L2_v20_13, the inventory costs have been divided by more than 6 (which corresponds to several M€).

The 2SA-m algorithm requires 90 seconds to output a solution, while lot-size and cover-size heuristics take less than a second and the deterministic approximation less than 10 seconds. The 2SA-m algorithm is thus slower, but note that 90 seconds to be run only once at the beginning of each week remains very short. Moreover, even with improved computers, the lot-size and cover-size heuristics and the deterministic approximation will not change their output (in all our experiments, Gurobi always found the optimal solution of the deterministic approximation). This is not the case for 2SA-m, which means that it would benefit from improved computational capacities.

Finally, we present in Fig. 7.3 the trade-off between service level and cost, for our approach and the three heuristics.

We see that our approach offers significantly better inventory costs than the ones obtained by the heuristics used by the clients of our partner. These two heuristics are however able to provide almost always very good fill rate service level – at the price of very high holding costs.



(a) Results for 30 runs of L2 with $v = 0.5$ and $\beta = 95\%$

(b) Average results for L2 for various values of desired service level β and $v = 0.5$

Figure 7.3: Representation of the numerical results of our proposed approach and three different heuristics.

Perspectives

A) Open theoretical questions

I gathered here some open theoretical questions I hope to contribute to.

TFDP convergence polynomial in the horizon

In my opinion, the asymptotic convergence result of [VL3] for SDDP algorithms was limited for various reasons: it did not provide any estimation on the number of iterations required to get an ε-solution, made the finitely supported noise assumption, and argued that the algorithm would randomly select every possible scenario an infinite number of times. The first bounds on iteration number were given in 2019 ([Lan20, ZS22]) and we extended the convergence theory to non-finitely supported noise in [VL16].

However, these convergence analyses first consider a specific deterministic node selection which provides an iteration bound that is polynomial in the horizon T. To extend the analysis to random sampling, we consider the probability of exactly selecting a single scenario (or a joint event for the non-finitely supported case), which yields a bound exponential in the horizon.

Open question: *can we prove a bound, polynomial in the horizon T, on the expected number of iterations required for a TFDP algorithm, with random node selection, to return an ε-solution to a (Lipchitz) MSP problem?*

A primal-dual SDDP algorithm

The SDDP algorithm is an extension of Kelley's cutting plane algorithm to the stochastic multistage case. Kelley's algorithm is known to be slow and unstable and has been greatly improved by adding some regularization scheme, resulting for example in the bundle methods. SDDP is also known to be slow, both in theory and practice. More precisely, we numerically observe that, after some time, the algorithm makes almost no more progress.

Therefore, adding a regularization procedure is an active idea in the community, and multiple propositions have been made recently ([VAdOS19, BFFdO20]), mostly relying on level-set regularization which requires upper bounds.

Open question: *can we design a provably faster SDDP algorithm that would run both a primal and dual SDDP (see Chapter 4), using the value of one to compute upper bounds for the other?*

Multi-scale MSLPs

The MSP setting considered in this manuscript tackles decisions taken at stage t that only impact the system through its next stage state x_{t+1}. However, there is a large class of problems where strategical or design decisions (*e.g.,* size or number of batteries bought in a microgrid) taken at stage $t = 0$ impact the system on all stages, while still having operational decisions that have an impact limited to one stage. More generally, there are problems with investment decisions taken, say every year, and operational decisions taken every week, that we call multiscale problems.

In the convex setting, using envelope theorems, we can obtain subgradients of the value function with respect to the strategical decisions, thus paving the way to cutting plane algorithms. However, these

subgradients are given as an expectation over the whole horizon of the system, which makes the exact computation intractable.

Open question: *can we design an efficient multiscale SDDP algorithm that allows for statistically estimated subgradients instead of exact ones?*

B) New applications and challenges

I present here some mathematical challenges stemming from industrial applications.

MSLP with a few binary variables: application to industrial microgrids

Due to the environmental crisis, the world of energy is fast changing. More renewable energies need to be integrated into the energy mix, in particular at a local, decentralized level to satisfy local consumption. Industrial microgrids are industrial complexes that mix energy system storage, local production (renewable or not), and potentially adjustable industrial demands. They can be connected to the grid, or isolated.

Some of these problems can be cast as MSLPs with a few binary variables representing semi-continuous variables, minimal uptime of downtime, maximum number of start-ups or shared resources constraints. Current TFDP algorithms are either dedicated to continuous variables or often slowly converging. I believe that there is ground for improvement for heuristics of exact methods based on SDDP, coupled with branching methodology, and exploiting specific structures of the problems.

This path of research is motivated by contracts with Metron Energy and TotalEnergies. The first partnership, funding the Ph.D. thesis of Zoé Fornier, is concerned with providing a central digital solution for industrial complexes. This solution shall jointly optimize the production planning and energy procurement of the complex. The second partnership, funding the Ph.D. thesis of Vitor Luiz Pinto Pereira, is concerned only with energy procurement planning, but for an isolated system where demand has to be satisfied for security reasons.

Challenges in supply chain

The supply chain is a vast area of challenges for optimization under uncertainty.

In addition to the lot-sizing problem presented in Chapter 7, we also discussed during Étienne thesis, a multi-sourcing problem, which decide to allocate production capacity to various center, balancing cost, stock level and flexibility of the supply chain. The proposed approach was a two-stage model with a service level constraint modeled as an AVAR constraint.

I believe that there is more work to be done on the subject, maybe leveraging multi-objective tools, to design a supply chain that simultaneously optimizes costs, stock, resiliency and sustainability. This is motivated by a contract with the Chair Supply Chain of Tomorrow funded by 4 companies (CDiscount, Louis Vuitton, Michelin and Renault), that funds the Post Doctorate internship of Carlos Moreno.

Another interesting challenge comes from Vehicle Routing Problems. This is motivated by a research contract, carried out in collaboration with Axel Parmentier, that aims at optimizing a large-scale (multi-depot, multi-item, split delivery) inventory routing problem as part of the backward logistics of Renault. In particular, Renault is using partners to deliver their products, and wants to contract some routes in advance. This constitutes a two-stage problem, where the recourse problem in itself is numerically challenging. As such, classical two-stage approaches fail, and we hope to leverage recent advances in structured machine learning to develop a heuristic method.

Bibliography

[ACdC20] Shabbir Ahmed, Filipe Goulart Cabral, and Bernardo Freitas Paulo da Costa. Stochastic lipschitz dynamic programming. *Mathematical Programming*, pages 1–39, 2020.

[AD54] Kenneth J Arrow and Gerard Debreu. Existence of an equilibrium for a competitive economy. *Econometrica: Journal of the Econometric Society*, pages 265–290, 1954.

[ADEH99] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.

[AP18] Tsvetan Asamov and Warren B Powell. Regularized decomposition of high-dimensional multistage stochastic programs with markov uncertainty. *SIAM Journal on Optimization*, 28(1):575–595, 2018.

[Att84] Hedy Attouch. *Variational convergence for functions and operators*, volume 1. Pitman Advanced Pub. Program, 1984.

[BCG10] Kengy Barty, Pierre Carpentier, and Pierre Girardeau. Decomposition of large-scale stochastic optimal control problems. *RAIRO-Operations Research*, 44(3):167–183, 2010.

[BDZ17] Regan Baucke, Anthony Downward, and Golbon Zakeri. A deterministic algorithm for solving multistage stochastic programming problems. *Optimization Online*, pages 1–25, 2017.

[BDZ18] Regan Baucke, Anthony Downward, and Golbon Zakeri. A deterministic algorithm for solving multistage stochastic minimax dynamic programmes. *Optimization Online*, 2018.

[Ber12] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.

[BFFdO20] Felipe Beltrán, Erlon C Finardi, Guilherme M Fredo, and Welington de Oliveira. Improving the performance of the stochastic dual dynamic programming algorithm using chebyshev centers. *Optimization and Engineering*, pages 1–22, 2020.

[Bir85] John R Birge. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations research*, 33(5):989–1007, 1985.

[BL11] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[BLG+20] Gilles Bareilles, Yassine Laguel, Dmitry Grishchenko, Franck Iutzeler, and Jérôme Malick. Randomized progressive hedging methods for multi-stage stochastic programming. *Annals of Operations Research*, 295(2):535–560, 2020.

[BS92] Louis J Billera and Bernd Sturmfels. Fiber polytopes. *Annals of Mathematics*, pages 527–549, 1992.

[CCDLP20] Pierre Carpentier, Jean-Philippe Chancelier, Michel De Lara, and François Pacaud. Mixed spatial and temporal decompositions for large-scale multistage stochastic optimization problems. *Journal of Optimization Theory and Applications*, 186(3):985–1005, 2020.

[Com18] Patrick L Combettes. Perspective functions: Properties, constructions, and examples. *Set-Valued and Variational Analysis*, 26(2):247–264, 2018.

[DDB20] Anthony Downward, Oscar Dowson, and Regan Baucke. Stochastic dual dynamic programming with stagewise-dependent objective uncertainty. *Operations Research Letters*, 48(1):33–39, 2020.

[DLRS10] Jesús A De Loera, Jörg Rambau, and Francisco Santos. *Triangulations Structures for algorithms and applications*. Springer, 2010.

[dMdES17] Gauthier de Maere d'Aertrycke, Andreas Ehrenmann, and Yves Smeers. Investment with incomplete markets for risk: The need for long-term contracts. *Energy Policy*, 105:571–583, 2017.

[dSG18] Etienne Gaillard de Saint-Germain. *Balancing cost and flexibility in supply chain*. PhD thesis, Université Paris-Est, 2018.

[ES11] Andreas Ehrenmann and Yves Smeers. Generation capacity expansion in a risky environment: a stochastic equilibrium analysis. *Operations Research*, 59(6):1332–1346, 2011.

[FDJM09] Michael C Ferris, Steven P Dirkse, Jan-H Jagla, and Alexander Meeraus. An extended mathematical programming framework. *Computers & Chemical Engineering*, 33(12):1973–1982, 2009.

[FM00] Michael C Ferris and Todd S Munson. Complementarity problems in GAMS and the PATH solver. *Journal of Economic Dynamics and Control*, 24(2):165–188, 2000.

[For22] Maël Forcier. *Multistage stochastic optimization and polyhedral geometry*. PhD thesis, École des Ponts, 2022.

[GLT20] Vincent Guigues, Migual A Lejeune, and Wajdi Tekaya. Regularized stochastic dual dynamic programming for convex nonlinear optimization problems. *Optimization and Engineering*, 21(3):1133–1165, 2020.

[GSC19] Vincent Guigues, Alexander Shapiro, and Yi Cheng. Duality and sensitivity analysis of multistage linear stochastic programs. *arXiv preprint arXiv:1911.07080*, 2019.

[GSC21] Vincent Guigues, Alexander Shapiro, and Yi Cheng. Risk-averse stochastic optimal control: an efficiently computable statistical upper bound. *arXiv preprint arXiv:2112.09757*, 2021.

[GTW19] Angelos Georghiou, Angelos Tsoukalas, and Wolfram Wiesemann. Robust dual dynamic programming. *Operations Research*, 67(3):813–830, 2019.

[Gui16] Vincent Guigues. Convergence analysis of sampling-based decomposition methods for risk-averse multistage stochastic convex programs. *SIAM Journal on Optimization*, 26(4):2468–2494, 2016.

[Gui20] Vincent Guigues. Inexact cuts in stochastic dual dynamic programming. *SIAM Journal on Optimization*, 30(1):407–438, 2020.

[HdMDMF11] Tito Homem-de Mello, Vitor L De Matos, and Erlon C Finardi. Sampling strategies and stopping criteria for stochastic dual dynamic programming: a case study in long-term hydrothermal scheduling. *Energy Systems*, 2(1):1–31, 2011.

[HKW16] Grani A Hanasusanto, Daniel Kuhn, and Wolfram Wiesemann. A comment on "computational complexity of stochastic programming problems". *Mathematical Programming*, 159(1-2):557–569, 2016.

[KF09] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[KM15] Václav Kozmík and David P Morton. Evaluating policies in risk-averse multi-stage stochastic programming. *Mathematical Programming*, 152(1):275–300, 2015.

[Kud74] H. Kudo. A note on the strong convergence of σ-algebras. *The Annals of Probability*, 2(1):76–83, 1974.

[Kuh06] Daniel Kuhn. *Generalized bounds for convex multistage stochastic programs*, volume 548. Springer Science & Business Media, 2006.

[Kuh08] Daniel Kuhn. Aggregation and discretization in multistage stochastic programming. *Mathematical Programming*, 113(1):61–94, 2008.

[Lan20] Guanghui Lan. Complexity of stochastic dual dynamic programming. *Mathematical Programming*, pages 1–38, 2020.

[Law91] Jim Lawrence. Polytope volume computation. *Mathematics of computation*, 57(195):259–271, 1991.

[LJ83] Hendrik W Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of operations research*, 8(4):538–548, 1983.

[LR08] Shu Lu and Stephen M Robinson. Normal fans of polyhedral convex sets. *Set-Valued Analysis*, 16(2-3):281–305, 2008.

[MAB14] Francesca Maggioni, Elisabetta Allevi, and Marida Bertocchi. Bounds in multistage linear stochastic programming. *Journal of Optimization Theory and Applications*, 163(1):200–229, 2014.

[MP16] Francesca Maggioni and Georg Ch Pflug. Bounds and approximations for multistage stochastic programs. *SIAM Journal on Optimization*, 26(1):831–855, 2016.

[OA16] Frans A Oliehoek and Christopher Amato. *A concise introduction to decentralized POMDPs*. Springer, 2016.

[OR99] Włodzimierz Ogryczak and Andrzej Ruszczyński. From stochastic dominance to mean-risk models: Semideviations as risk measures. *European journal of operational research*, 116(1):33–50, 1999.

[OR01] Włodzimierz Ogryczak and Andrzej Ruszczyński. On consistency of stochastic dominance and mean–semideviation models. *Mathematical Programming*, 89(2):217–232, 2001.

[PDLCC21] François Pacaud, Michel De Lara, Jean-Philippe Chancelier, and Pierre Carpentier. Distributed multistage optimization of large-scale microgrids under stochasticity. *IEEE Transactions on Power Systems*, 37(1):204–211, 2021.

[PdMF13] Andy Philpott, Vitor de Matos, and Erlon Finardi. On solving multistage stochastic programs with coherent risk measures. *Operations Research*, 61(4):957–970, 2013.

[PFW16] A. Philpott, M. Ferris, and R. Wets. Equilibrium, uncertainty and risk in hydro-thermal electricity systems. *Mathematical Programming*, 157(2):483–513, 2016.

[PG08] Andrew B Philpott and Ziming Guan. On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36(4):450–455, 2008.

[Pic98] Laurent Piccinini. A new version of the multivalued fatou lemma. *Journal of Applied Analysis*, 4(2), 1998.

[Pow14] Warren B. Powell. *Clearing the Jungle of Stochastic Optimization*, chapter Chapter 4, pages 109–137. 2014.

[Pow19] Warren B Powell. A unified framework for stochastic optimization. *European Journal of Operational Research*, 275(3):795–821, 2019.

[PP91] Mario VF Pereira and Leontina MVG Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical programming*, 52(1-3):359–375, 1991.

[PP12] Georg Pflug and Alois Pichler. On dynamic decomposition of multistage stochastic programs. *Optimization Online preprint*, 2(11), 2012.

[PZP10] Geoffrey Pritchard, Golbon Zakeri, and Andy Philpott. A single-settlement, energy-only electric power market for unpredictable and intermittent participants. *Operations Research*, 58:1210–1219, 2010.

[QGKS22] Franco Quezada, Céline Gicquel, and Safia Kedad-Sidhoum. A stochastic dual dynamic integer programming based approach for remanufacturing planning under uncertainty. *International Journal of Production Research*, pages 1–21, 2022.

[Ram96] Jörg Rambau. *Polyhedral subdivisions and projections of polytopes*. Phd thesis, TU Berlin, 1996.

[RBHdM19] Hamed Rahimian, Güzin Bayraksan, and Tito Homem-de Mello. Identifying effective scenarios in distributionally robust stochastic programs with total variation distance. *Mathematical Programming*, 173(1):393–430, 2019.

[RPM21] Cristian Ramirez-Pico and Eduardo Moreno. Generalized adaptive partition-based method for two-stage stochastic linear programs with fixed recourse. *Mathematical Programming*, pages 1–20, 2021.

[RS06a] Andrzej Ruszczyński and Alexander Shapiro. Conditional risk mappings. *Mathematics of operations research*, 31(3):544–561, 2006.

[RS06b] Andrzej Ruszczyński and Alexander Shapiro. Optimization of convex risk functions. *Mathematics of operations research*, 31(3):433–452, 2006.

[RS15] Daniel Ralph and Yves Smeers. Risk trading and endogenous probabilities in investment equilibria. *SIAM Journal on Optimization*, 25(4):2589–2611, 2015.

[RU+00] R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[Rus97] Andrzej Ruszczyński. Decomposition methods in stochastic programming. *Mathematical programming*, 79(1):333–353, 1997.

[RW91] R Tyrrell Rockafellar and Roger J-B Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research*, 16(1):119–147, 1991.

[RW98] Tyrell Rockafellar and Roger Wets. *Variational Analysis*, volume 317. Springer Verlag, 1998.

[RZ96] Jörg Rambau and Günter M Ziegler. Projections of polytopes and the generalized baues conjecture. *Discrete & Computational Geometry*, 16(3):215–237, 1996.

[SAB+20] Adrien Seguret, Clémence Alasseur, J Frédéric Bonnans, Antonio De Paola, Nadia Oudjane, and Vincenzo Trovato. Decomposition of high dimensional aggregative stochastic control problems. *arXiv preprint arXiv:2008.09827*, 2020.

[Sam41] Paul A Samuelson. The stability of equilibrium: comparative statics and dynamics. *Econometrica: Journal of the Econometric Society*, pages 97–120, 1941.

[SDR14] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

[Sha11] Alexander Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72, 2011.

[SL15] Yongjia Song and James Luedtke. An adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse. *SIAM Journal on Optimization*, 25(3):1344–1367, 2015.

[Spa12] Matthijs TJ Spaan. Partially observable markov decision processes. In *Reinforcement Learning*, pages 387–414. Springer, 2012.

[SS22] Murwan Siddig and Yongjia Song. Adaptive partition-based sddp algorithms for multistage stochastic linear programming with fixed recourse. *Computational Optimization and Applications*, 81(1):201–250, 2022.

[Uza60] Hirofumi Uzawa. Walras' tatonnement in the theory of exchange. *The Review of Economic Studies*, 27(3):182–194, 1960.

[VAdOS19] Wim Van Ackooij, Welington de Oliveira, and Yongjia Song. On level regularization with normal solutions in decomposition methods for multistage stochastic programming problems. *Computational Optimization and Applications*, 74(1):1–42, 2019.

[VSW69] Richard M Van Slyke and Roger Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663, 1969.

[Wet89] Roger JB Wets. The aggregation principle in scenario analysis and stochastic optimization. In *Algorithms and model formulations in mathematical programming*, pages 91–113. Springer, 1989.

[WF07] S. Wong and J. Fuller. Pricing energy and reserves using stochastic optimization in an alternative electricity market. *IEEE Transactions on Power Systems*, 22(2):631–638, 2007.

[WW11] Jean-Paul Watson and David L Woodruff. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science*, 8(4):355–370, 2011.

[ZPBB16] G. Zakeri, G. Pritchard, M. Bjorndal, and E. Bjorndal. Pricing wind: A revenue adequate cost recovering uniform price for electricity markets with intermittent generation. *NHH Dept. of Business and Management Science*, 2016.

[ZS22] Shixuan Zhang and Xu Andy Sun. Stochastic dual dynamic programming for multistage stochastic mixed-integer nonlinear optimization. *Mathematical Programming*, pages 1–51, 2022.