Résolution de grands problèmes en optimisation stochastique dynamique et synthèse de lois de commande

Pierre Girardeau

UNIVERSITÉ PARIS-EST

Thèse

présentée pour l'obtention du titre de

Docteur de l'Université Paris-Est

Spécialité : Mathématiques

par

Pierre GIRARDEAU

Résolution de grands problèmes en optimisation stochastique dynamique et synthèse de lois de commande

---

Soutenance le 17 décembre 2010 devant le jury composé de :

Rapporteurs :	Jean-Pierre Quadrat	INRIA Paris-Rocquencourt
	Nizar Touzi	École Polytechnique
Examinateurs :	Kengy Barty	EDF R&D
	Andrew Philpott	University of Auckland
	Felisa Vázquez-Abad	Hunter College, New York
Directeurs de thèse :	Pierre Carpentier	ENSTA-ParisTech
	Guy Cohen	École des Ponts-ParisTech

ENSTA ParisTech — École des Ponts ParisTech — EDF R&D

Cette thèse a été effectuée conjointement à l'Unité de Mathématiques Appliquées de l'ENSTA-ParisTech, au CERMICS de l'École des Ponts-ParisTech, et dans le département OSIRIS d'EDF R&D sous la forme d'un contrat CIFRE.

Cette thèse a été rédigée à l'aide du logiciel libre de mise en forme LaTeX, ainsi que des précieux conseils proposés par Mori (2008) au sujet de la rédaction d'une thèse.

Résumé

Le travail présenté ici s'intéresse à la résolution numérique de problèmes de commande optimale stochastique de grande taille. Nous considérons un système dynamique, sur un horizon de temps discret et fini, pouvant être influencé par des bruits exogènes et par des actions prises par le décideur. L'objectif est de contrôler ce système de sorte à minimiser une certaine fonction objectif, qui dépend de l'évolution du système sur tout l'horizon. Nous supposons qu'à chaque instant des observations sont faites sur le système, et éventuellement gardées en mémoire. Il est généralement profitable, pour le décideur, de prendre en compte ces observations dans le choix des actions futures. Ainsi sommes-nous à la recherche de stratégies, ou encore de lois de commandes, plutôt que de simples décisions. Il s'agit de fonctions qui à tout instant et à toute observation possible du système associent une décision à prendre.

Ce manuscrit présente trois contributions. La première concerne la convergence de méthodes numériques basées sur des scénarios. Nous comparons l'utilisation de méthodes basées sur les arbres de scénarios aux méthodes particulaires. Les premières ont été largement étudiées au sein de la communauté "Programmation Stochastique". Des développements récents, tant théoriques que numériques, montrent que cette méthodologie est mal adaptée aux problèmes à plusieurs pas de temps. Nous expliquons ici en détails d'où provient ce défaut et montrons qu'il ne peut être attribué à l'usage de scénarios en tant que tel, mais plutôt à la structure d'arbre. En effet, nous montrons sur des exemples numériques que les méthodes particulaires, plus récemment développées et utilisant également des scénarios, ont un meilleur comportement même avec un grand nombre de pas de temps.

La deuxième contribution part du constat que, même à l'aide des méthodes particulaires, nous faisons toujours face à ce qui est couramment appelé, en commande optimale, la *malédiction de la dimension*. Lorsque l'état servant à résumer le système est de trop grande taille, on ne sait pas trouver directement, de manière satisfaisante, des stratégies optimales. Pour une classe de systèmes, dits décomposables, nous adaptons des résultats bien connus dans le cas déterministe, portant sur la décomposition de grands systèmes, au cas stochastique. L'application n'est pas directe et nécessite notamment l'usage d'outils statistiques sophistiqués afin de pouvoir utiliser la variable duale qui, dans le cas qui nous intéresse, est un processus stochastique. Nous proposons un algorithme original appelé *Dual Approximate Dynamic Programming* (DADP) et étudions sa convergence. Nous appliquons de plus cet algorithme à un problème réaliste de gestion de production électrique sur un horizon pluri-annuel.

La troisième contribution de la thèse s'intéresse à une propriété structurelle des problèmes de commande optimale stochastique : la question de la consistance dynamique d'une suite de problèmes de décision au cours du temps. Notre but est d'établir un lien entre la notion de consistance dynamique, que nous définissons de manière informelle dans le dernier chapitre, et le concept de variable d'état, qui est central dans le contexte de la commande optimale. Le travail présenté est original au sens suivant. Nous montrons que, pour une large classe de modèles d'optimisation stochastique n'étant pas a priori consistants dynamiquement, on peut retrouver la consistance dynamique quitte à étendre la structure d'état du système.

Abstract

This work intends to provide resolution methods for Stochastic Optimal Control (SOC) problems. We consider a dynamical system on a discrete and finite horizon, which is influenced by exogenous noises and actions of a decision maker. The aim is to minimize a given function of the system's behaviour over the whole time horizon. We suppose that at every instant the decision maker is able to make observations on the system and keep some in memory. Since it is generally profitable to take these observations into account in order to draw further actions, we aim to design decision rules rather than simple decisions. Such rules associate to every instant and every possible observation of the system a decision to make.

The present manuscript presents three main contributions. The first concerns the study of scenario-based solving methods for SOC problems. We compare the use of the so-called scenario trees technique to the particle methods. The first one has been widely studied among the Stochastic Programming community and has been somehow popular in applications; however recent developments showed numerically as well as theoretically that this methodology behaves poorly when the number of th problem's time steps grows. We explain this fact in details and show that this negative feature is not to be attributed to the scenario setting, but rather to the use of the tree structure. Indeed, we show using numerical examples how the particle method – which is a newly developed variational technique also based on scenarios – behaves in a better way even when we deal with a large number of time steps.

The second contribution starts from the observation that, even with particle methods, we are still facing somehow the curse of dimensionality. In other words, decision rules intrisically suffer from the dimension of their domain, e.g. observations or state in the Dynamic Programming framework. For a certain class of systems, namely decomposable systems, we adapt results concerning the decomposition of large-scale systems which are well known in the deterministic case to the SOC case. The application is not straightforward and requires some statistical analysis for the dual variable, which is a stochastic process in our context. We propose an innovating algorithm called Dual Approximate Dynamic Programming (DADP) and study its convergence. We also apply DADP to a real-life power management problem.

The third contribution concerns a rather structural property for SOC problems: the question of dynamic consistency for a sequence of decision making problems over time. Our aim is to establish a link between the notion of time consistency, that we loosely define in the last chapter, and the central concept of state structure within optimal control. This contribution is original in the following sense: many works in the literature aim to find optimization models which somehow preserve the "natural" time consistency property for the sequence of decision making problems. On the contrary, we show for a broad class of SOC problems which are not a priori time-consistent, that it is possible to regain this property by simply extending the state structure of the model.

Remerciements

Mes premiers remerciements vont aux personnes qui ont accepté de participer à mon jury de thèse. Les enseignements que j'ai eu la chance de recevoir de Jean-Pierre Quadrat m'ont été utiles tout au long de ces trois ans. Ses travaux en commande optimale et autour de la programmation dynamique m'ont souvent, au cours de cette période de thèse, servi d'inspiration. Je voudrais aussi remercier Nizar Touzi pour sa sympathie et pour l'intérêt qu'il a pu porter à nos recherches lors des séminaires et conférences au cours desquels j'ai eu le plaisir de le rencontrer. Il serait évidemment impensable de ne pas remercier mon collègue et néanmoins ami Kengy Barty, qui a beaucoup fait pour que cette thèse se passe dans les meilleures conditions et avec qui j'ai eu la chance, depuis maintenant cinq ans, de travailler dans une ambiance sereine, studieuse et amicale. J'espère que nous continuerons longtemps cette "collaboration". J'ai eu l'occasion de rencontrer Andrew Philpott en 2008 lors de la conférence ISMP à Chicago. Suite à nos discussions, il s'est rapidement montré intéressé et encourageant à l'égard de nos travaux. J'aimerais ici le remercier chaleureusement, non seulement pour avoir bien voulu faire partie de mon jury de thèse, mais aussi pour me donner l'occasion de travailler avec lui dans l'année et demie qui arrive. Je remercie aussi Felisa Vázquez-Abad, entre autres pour les discussions tant scientifiques qu'amicales que nous avons pu avoir, que ce soit en France ou en Australie, lorsqu'elle m'a accueilli en 2008 à l'Université de Melbourne. *Last but not least*, toute ma gratitude va à mes directeurs de thèse, Pierre Carpentier et Guy Cohen. J'ai eu le privilège d'être activement suivi par deux scientifiques passionnés et passionnants qui ont su tantôt m'encourager, tantôt me mettre au défi, et ce toujours dans un climat à la fois studieux et sympathique.

Au cours de ces trois années, j'ai passé la plupart de mon temps au sein du département Optimisation, Simulation, Risque et Statistique (OSIRIS) d'EDF R&D. Quand je repense aux débuts, mes premières pensées vont aux trois mousquetaires qui avaient encadré mon stage de césure dans ce département et dont l'exemple m'avait incité à faire le choix de poursuivre en thèse : Kengy Barty que je n'ai pas peur de remercier ainsi deux fois, Cyrille Strugarek, mon grand frère spirituel, et Jean-Sébastien Roy, malheureusement parti trop tôt. Je leur adresse mes plus sincères remerciements. Par ailleurs, je tiens à remercier les trois chefs de groupe qui m'ont accueilli, que ce soit au cours du stage ou de la thèse : René Aïd, Luciano Leal de Sousa et Sandrine Charousset, ainsi que Yannick Jacquemart, chef du département OSIRIS, pour avoir cru à ce projet et pour me permettre de continuer encore l'aventure. Je me suis dit que si je commençais à dresser la liste des collègues avec qui j'ai eu la chance de partager parfois un café et souvent bien plus, je ferais certainement trop d'oublis. Alors je remercie tout à la fois les Sfaxiens, la Beyrouthoise, les Nantais, le Bratislavien, le Martiniquais, les Orléanais, les Tunisois (et la Tunisoise), les Angevins bien sûr, peut-être même les Parisiens et je m'arrête puisque c'est là que les Athéniens s'atteignirent.

Je voudrais aussi remercier les enseignants-chercheurs et les doctorants de l'ENSTA et

du CERMICS que j'ai trop peu croisé pendant la thèse. En particulier, j'adresse un grand merci à Michel De Lara pour ses commentaires et opinions constructives à propos de mes travaux de thèse et de leur présentation, ainsi qu'à Jean-Philippe Chancelier tant pour son aide en informatique que pour les agréables et diverses discussions que nous avons eues pendant ces trois ans.

Toute ma gratitude va à Jean-Charles Gilbert ainsi qu'à Hasnaa Zidani pour m'avoir permis d'encadrer des travaux dirigés au sein de leurs cours respectifs.

Aux éléments qui ont permis que cette thèse se déroule dans les meilleures conditions viennent s'ajouter sans hésitation les amis, en premier lieu ceux du Cent-Quinze, sans qui je n'aurais souvent pas eu la force de continuer. Leur présence à la soutenance m'a fait le plus grand plaisir. Sans chercher à être exhaustif, je me dois de remercier le "noyau dur" : mes deux colocataires Bébert et Seb, le cousin, Alex, Dédelle, Fabi, la Monnier, la mèche, le Mignon, et *What else ?*

Finalement, j'adresse un grand et très sincère merci à ma famille (même à mon beau-frère) qui n'a jamais douté que ce projet aboutisse, même lorsque ma confiance s'ébranlait. J'ai envie de leur dire que parmi les choses qui ont été nécessaires à la réussite de cette thèse, il y a sans aucun doute un certain nombre de cours de mathématiques, mais il y a avant tout la richesse de ce qu'ils ont su me transmettre.

Avant-propos

À la lecture du titre de cette thèse et des mots barbares qui le composent, je félicite les courageux novices qui oseront ouvrir ce document. Je vais m'efforcer de justifier succinctement l'usage de ces termes, car chacun y a sa place.

Le mot qui surprend le plus le lecteur non familier des probabilités est sans doute *stochastique*. Je vais me garder d'en donner l'étymologie puisqu'elle est déjà élégamment énoncée dans la thèse de Cyrille Strugarek en avant-propos ; c'est d'ailleurs ce qui m'a donné l'envie d'écrire ces quelques lignes. Je me contenterai donc d'en donner la définition du dictionnaire (Larousse, 2010) : "Se dit de phénomènes qui, partiellement, relèvent du hasard et qui font l'objet d'une analyse statistique." On trouve également que *stochastique* a pour synonyme *aléatoire*. Dès lors, on pourrait croire que l'usage du premier terme plutôt que du second a pour seul but de "faire savant". Mais ce n'en est pourtant pas la raison.

Pour comprendre, il suffit d'ajouter le mot *optimisation*. L'optimisation est un domaine des mathématiques où l'on s'intéresse à la minimisation (ou à la maximisation) d'un certain objectif, tel qu'une valeur économique ou encore une énergie. Ce sujet est à la fois très ancien – les premiers problèmes d'optimisation remontent à Euclide – et relativement jeune – le développement des méthodes numériques telles que la programmation linéaire a connu un réel essor depuis la seconde moitié du 20^{e} siècle. On peut penser, pour se faire une idée, au problème de la recherche d'une route en temps minimal reliant deux points d'une carte. Certains paramètres du problème peuvent être incertains – il est possible que l'on rencontre par exemple des embouteillages sur la route – et l'optimisation va alors consister à rechercher le meilleur compromis entre tous les aléas possibles. Parler d'optimisation aléatoire laisserait croire que l'on va se résoudre à tirer la route à pile ou face, ce qui est généralement loin d'être optimal. On préfère donc parler d'optimisation stochastique.

En présence d'incertain, l'optimiseur (ou décideur) va souvent bénéficier d'informations sur le système à optimiser qui arriveront de manière *dynamique*, c'est-à-dire au fur et à mesure que le temps passe – on apprend par exemple au fur et à mesure que l'on teste les routes celles qui sont le plus sujettes aux embouteillages. La difficulté du problème d'optimisation sera alors étroitement liée à la quantité d'information qui est nécessaire à la prise de décision optimale. On parle de *grand problème* lorsque cette quantité d'information est trop importante pour employer brutalement les techniques classiques de résolution.

Pour finir, on a voulu insister, à travers l'expression *synthèse de lois de commande*, sur le fait que nous ne cherchons pas seulement à évaluer le coût optimal du système – le temps associé à la route optimale – mais surtout la *stratégie* (ou loi de commande) permettant d'y parvenir.

En espérant que la rédaction de cet avant-propos a permis de maximiser la probabilité que vous continuiez votre lecture.

Table des matières

Table des figures

Liste des symboles

$\mathbb{N}$	ensemble des entiers naturels
$\mathbb{R}$	ensemble des nombres réels
$\preceq$	est mesurable par rapport à
s.c.	sous les contraintes
t.q.	tel que
$:=$	est égal, par définition, à
$\boldsymbol{X}$	(en lettres **grasses**) variable aléatoire $\boldsymbol{X}$
$\mathbb{P}(A)$	probabilité de l'évènement A
$\mathbb{E}$	opérateur d'espérance
$\mathbb{E}\left(\boldsymbol{Y} \mid \boldsymbol{X}\right)$	espérance conditionnelle de $\boldsymbol{Y}$ sachant $\boldsymbol{X}$
$\mathbf{1}_A$	fonction indicatrice de l'ensemble A : $\mathbf{1}_A(x) = 1$ si $x \in A$; $\mathbf{1}_A(x) = 0$ sinon.
χ_A	fonction caractéristique de l'ensemble A : $\chi_A(x) = 0$ si $x \in A$; $\chi_A(x) = +\infty$ sinon.
i.i.d.	indépendantes et identiquement distribuées
$\sigma(\boldsymbol{X})$	tribu engendrée par la variable aléatoire $\boldsymbol{X}$
$\partial f(x)$	sous-différentiel de la fonction f en x
$\nabla f(x)$	gradient de la fonction f en x
Π_A	opérateur de projection sur l'ensemble A

Chapitre 1

Préliminaires

> Il entre dans toutes les actions humaines plus de hasard que de décision.
>
> André Gide (1869-1951)

Ce chapitre introductif a pour objet de présenter les concepts qui seront utiles tout au long de ce mémoire. On présente au §1.1 les types de problèmes auxquels nous nous intéressons par la suite. Au §1.2, nous décrivons brièvement quelques méthodes de résolution classiques sur lesquelles nous reviendrons par la suite. Enfin, au §1.3, nous présentons le plan de ce document en insistant sur les contributions apportées lors de cette thèse.

1.1 Optimisation dans l'incertain

Nous introduisons ici les concepts mathématiques permettant de modéliser des problèmes d'optimisation dans lesquels interviennent des aléas : on parle de problèmes d'optimisation *stochastique*. Les aléas peuvent être présents à la fois dans le critère à optimiser et dans les contraintes du problème. Nous présentons dans un premier temps les deux grandes classes de problèmes de ce type qui sont : d'une part les problèmes en boucle ouverte, pour lesquels les décisions peuvent être prises a priori, avant que le système n'évolue, d'autre part les problèmes en boucle fermée où les décisions sont prises au cours du temps et dépendent des observations faites sur le système au cours de son évolution.

Cette présentation est inspirée de celles faites par Barty (2004), Strugarek (2006) et Dallagi (2007) dans leurs thèses. Dans toute la suite, les variables aléatoires seront notées en caractères gras.

1.1.1 Problème général

Soit $\boldsymbol{U}$ (respectivemement $\boldsymbol{W}$) une variable aléatoire définie sur un espace probabilisé [1] $(\Omega, \mathcal{A}, \mathbb{P})$ à valeurs dans un espace de Hilbert $\mathbb{U}$ (respectivement $\mathbb{W}$). On notera également $\mathcal{U} := L^2(\Omega, \mathcal{A}, \mathbb{P}\,; \mathbb{U})$ et $\mathcal{W} := L^2(\Omega, \mathcal{A}, \mathbb{P}\,; \mathbb{W})$. Nous considérons des espaces de variables aléatoires de carré intégrable car ce sont des espaces de Hilbert. Or, nous cherchons à définir des algorithmes d'optimisation faisant souvent usage de la notion de

1. On renvoie le lecteur à l'annexe B pour la définition des notions classiques de probabilités telles qu'une tribu ou un espace probabilisé.

gradient, qui se définit naturellement dans les espaces de Hilbert. Soit $j : \mathcal{U} \times \mathcal{W} \to \mathbb{R}$ une intégrande normale (voir Définition B.4). Un grand nombre de problèmes d'optimisation stochastique peuvent être formulés de la manière suivante :

$$\min_{\boldsymbol{U} \in \mathcal{U}^{\mathrm{ad}}} J(\boldsymbol{U}) := \mathbb{E}\left(j(\boldsymbol{U}, \boldsymbol{W})\right), \tag{1.1}$$

où l'ensemble admissible $\mathcal{U}^{\mathrm{ad}}$ est un sous-ensemble de $\mathcal{U}$. C'est sur cet ensemble que l'on fera porter les contraintes sur la commande $\boldsymbol{U}$.

Les problèmes qui nous intéressent sont donc des problèmes d'optimisation particuliers, où la variable d'optimisation est une variable aléatoire. Une autre particularité est que la fonction J n'est généralement pas connue de manière analytique et doit être obtenue à partir de j, par exemple à l'aide d'un grand nombre d'évaluations de la fonction j. Au lieu de tenter d'utiliser brutalement les outils issus de la programmation mathématique, on peut alors tirer profit de cette structure afin de limiter le nombre d'évaluations de cette fonction j.

Nous allons maintenant préciser la forme de l'ensemble admissible $\mathcal{U}^{\mathrm{ad}}$. Donnons-nous une tribu $\mathcal{G}$ sur Ω et une multi-application $\boldsymbol{\Gamma} : \Omega \rightrightarrows \mathcal{P}(\mathbb{U})$ qui soit $\mathcal{A}$-mesurable [2]. On distinguera souvent dans la suite deux types de contraintes :

– des contraintes dites ponctuelles ou presque-sûres :

$$\mathcal{U}^{\mathrm{ps}} := \{\boldsymbol{U} \in \mathcal{U}, \boldsymbol{U} \in \boldsymbol{\Gamma}, \mathbb{P}\text{-p.s.}\},$$

– et des contraintes de mesurabilité :

$$\mathcal{U}^{\mathrm{mes}} := \{\boldsymbol{U} \in \mathcal{U}, \boldsymbol{U} \text{ est } \mathcal{G}\text{-mesurable}\}. \tag{1.2}$$

À la place de "$\boldsymbol{U}$ est $\mathcal{G}$-mesurable", on notera souvent : $\boldsymbol{U} \preceq \mathcal{G}$. Notons que, par définition, $\mathcal{U}^{\mathrm{mes}}$ est un sous-espace vectoriel de $\mathcal{U}$.

On pose alors : $\mathcal{U}^{\mathrm{ad}} = \mathcal{U}^{\mathrm{ps}} \cap \mathcal{U}^{\mathrm{mes}}$.

Les contraintes ponctuelles servent à modéliser, pour toute réalisation de l'aléa, les contraintes habituelles recontrées en optimisation mathématique (des contraintes de bornes, par exemple). En revanche, les contraintes de mesurabilité sont particulières à l'optimisation stochastique. Chaque élément de la tribu $\mathcal{G}$ pouvant être interprété comme une information, elles représentent le fait que la décision peut dépendre d'observations sur le système. Nous explicitons par la suite plusieurs cas importants.

Remarque 1.1. Ce cadre est en fait suffisamment général pour recouvrir la plupart des problèmes qui nous intéressent ici. Il suffit de préciser le sens que l'on donne au terme "décision". Le cas de la boucle ouverte [3] est celui où rien n'est observé. La décision est alors la même quelle que soit la réalisation de l'aléa. Au contraire, pour un problème en boucle fermée à deux pas de temps, on prend une première décision sans observation préalable, puis un aléa se réalise et est observé. Sur la base de cette observation, on doit prendre une seconde décision. Les problèmes à plusieurs pas de temps rentrent également dans ce cadre. Ils nécessitent d'introduire une collection de contraintes d'information intermédiaires au cours du déroulement de l'expérience signifiant que l'observation du système arrive progressivement, et qu'une commande à un instant intermédiaire ne peut dépendre que de l'information disponible à cet instant.

2. Nous faisons référence aux travaux de Rockafellar et Wets (1998, Chapitre 14) pour la définition de la mesurabilité d'une multi-application.

3. Nous revenons sur les notions de "boucle ouverte" et de "boucle fermée" par la suite. Le lecteur qui n'est pas familier de ces notions peut, dans un premier temps, passer cette remarque.

1.1.2 Effet dual

De manière générale, la tribu $\mathcal{G}$ peut dépendre de la variable $\boldsymbol{U}$. On parle alors de problème avec effet dual, ou encore de structure d'information dynamique. La définition suivante donne plus de précisions à ce sujet.

Définition 1.1 (Absence d'effet dual/Information statique)**.** Le problème est dit en information statique si la tribu d'information $\mathcal{G}$ ne dépend pas de la commande $\boldsymbol{U}$. On dit alors qu'on a absence d'effet dual.

Dans le cas contraire, on dit qu'on est en présence d'effet dual. Ce type de problèmes est sensiblement plus complexe que les problèmes en information statique. Supposons par exemple que l'on soit dans le cas où $\mathcal{G} := \sigma(h(\boldsymbol{U}, \boldsymbol{W}))$, avec h une certaine fonction mesurable. La contrainte de mesurabilité du problème (1.1) peut alors s'écrire :

$$\boldsymbol{U} = \mathbb{E}\left(\boldsymbol{U} \mid h\left(\boldsymbol{U}, \boldsymbol{W}\right)\right), \qquad \mathbb{P}\text{-p.s.}$$

Alors, si on souhaite utiliser des techniques variationnelles pour résoudre le problème d'optimisation, on observe que l'on doit calculer un gradient par rapport à une variable $\boldsymbol{U}$ qui est présente dans le conditionnement d'une espérance conditionnelle. Or le calcul différentiel par rapport à un conditionnement est loin d'être trivial. Pour s'en convaincre, il suffit de considérer le cas $h(\boldsymbol{U}, \boldsymbol{W}) = \varepsilon \boldsymbol{U}$. Pour tout ε non nul, on a que $\mathbb{E}\left(\boldsymbol{U} \mid \varepsilon \boldsymbol{U}\right) = \boldsymbol{U}$. Or, pour $\varepsilon = 0$, on a que $\mathbb{E}\left(\boldsymbol{U} \mid \varepsilon \boldsymbol{U}\right) = \mathbb{E}\left(\boldsymbol{U}\right)$.

Ainsi, comme le montre Witsenhausen (1968) dans le cas d'un système linéaire quadratique gaussien, les propriétés de tels problèmes peuvent être assez éloignées des propriétés "habituelles". On pourra consulter l'article de Barty, Carpentier, Chancelier, Cohen, de Lara, et Guilbaud (2006) ou encore la thèse de Strugarek (2006, Chapitre II) pour des développements récents à ce sujet.

Nous nous concentrerons par la suite sur des problèmes en information statique.

1.1.3 Problèmes en boucle ouverte

Supposons maintenant que la tribu $\mathcal{G}$ ne dépend pas de $\boldsymbol{U}$. Il reste tout de même un certain nombre de possibilités pour $\mathcal{G}$. Nous citons maintenant la plus simple.

Définition 1.2 (Boucle ouverte, boucle fermée)**.** On dit que le problème (1.1) est en boucle ouverte si :

$$\mathcal{U}^{\text{mes}} \subset \left\{\boldsymbol{U} \in \mathcal{U}, \text{ tel que } \boldsymbol{U} \text{ est } \sigma\left\{\emptyset, \Omega\right\}\text{-mesurable}\right\}.$$

Dans le cas contraire, le problème est dit en boucle fermée.

Le cas de la boucle ouverte est donc celui où $\mathcal{G}$ est la tribu grossière $\sigma\{\emptyset, \Omega\}$. Les problèmes en boucle ouverte visent ainsi à modéliser le cas où la décision est prise sans aucune information. Autrement dit, toutes les décisions doivent être prises dès le tout premier instant, c'est-à-dire avant l'intervention de l'aléa. On cherche alors la décision réalisant le meilleur compromis entre les aléas.

À l'inverse, les problèmes en boucle fermée permettent de décrire des cas où le décideur est capable d'observer tout ou partie de l'aléa, et peut faire intervenir ces observations dans sa prise de décision. Nous nous intéressons plutôt par la suite aux problèmes en boucle fermée.

Cette terminologie “boucle ouverte-boucle fermée” provient de la communauté automaticienne dans laquelle on peut placer celle de la commande (ou du contrôle) stochastique, cette communauté ayant pour vocation de traiter de problèmes dynamiques (où intervient le temps comme évoqué ci-dessous). Plus récemment, la communauté *Stochastic Programming*, plutôt issue de celle du *Mathematical Programming* ayant au départ pour vocation de traiter de problèmes d'optimisation statiques, a redécouvert cette notion de “boucle fermée” (*feedback*) en introduisant diverses nouvelles terminologies (“avec recours”, “wait and see” par opposition à “here and now” pour la boucle ouverte).

Les problèmes en boucle ouverte sont généralement moins complexes que les problèmes en boucle fermée, notamment du fait de la nature des variables d'optimisation : dans le cas de la boucle ouverte, nous sommes à la recherche de décisions a priori, alors que dans le cas de la boucle fermée nous cherchons une décision pour chaque observation possible du système. De plus, on peut souvent utiliser des outils proches de ceux de l'optimisation déterministe, tels que le gradient stochastique (voir Robbins et Monro, 1951, Bertsekas et Tsitsiklis, 2000, ou encore Quadrat, Gousat, Hertz, et Viot, 1981, pour l'utilisation du gradient stochastique sur un problème d'investissement optimal) pour traiter les problèmes en boucle ouverte. Quand de tels problèmes sont de grande taille, on peut également adapter les méthodes de décomposition connues dans le cas déterministe (voir Cohen et Culioli, 1990).

1.1.4 Interprétation fonctionnelle

Nous montrons maintenant que le problème d'optimisation stochastique (1.1) est équivalent, sous certaines hypothèses, à un problème d'optimisation fonctionnelle. Nous avons essentiellement besoin qu'il existe une fonction d'observation h définie sur $\mathbb{W}$, à valeurs dans un espace de Hilbert $\mathbb{Y}$, mesurable, telle que nous puissions écrire la variable d'observation $\boldsymbol{Y}$ comme une fonction du bruit $\boldsymbol{W}$.

Proposition 1.3 (Équivalence entre problème stochastique et problème fonctionnel)**.** *Supposons qu'il existe une fonction mesurable* $h : \mathbb{W} \to \mathbb{Y}$ *ainsi qu'une multi-application mesurable* $C : \mathbb{W} \rightrightarrows \mathbb{U}$ *telles que* $\boldsymbol{Y} = h(\boldsymbol{W})$ *et* $\boldsymbol{\Gamma} = C(\boldsymbol{W})$, $\mathbb{P}$*-p.s. Alors le problème* (1.1) *est équivalent au problème :*

$$\min_{\phi \in \Phi^{\mathrm{ad}}} \quad \tilde{J}(\phi) := \mathbb{E}\left(j\left(\phi\left(\boldsymbol{W}\right), \boldsymbol{W}\right)\right),$$

où $\Phi^{\mathrm{ad}} := \Phi^{ps} \cap \Phi^{mes}$ *et :*

$$\begin{aligned} \Phi^{ps} &:= \left\{\phi \ : \mathbb{W} \to \mathbb{U}, \phi\left(\boldsymbol{W}\right) \in C\left(\boldsymbol{W}\right), \mathbb{P}\text{-}p.s.\right\}, \\ \Phi^{mes} &:= \left\{\phi \ : \mathbb{W} \to \mathbb{U}, \phi \preceq h\right\}. \end{aligned}$$

Démonstration. D'après l'expression (1.2), il existe une fonction mesurable $p : \mathbb{Y} \to \mathbb{U}$ telle que $\boldsymbol{U} = p \circ h(\boldsymbol{W})$ (voir Dellacherie et Meyer, 1975, Chapitre 1, p. 18, pour l'existence d'une telle fonction p). Notons $\phi := p \circ h$. On a alors $\phi \preceq h$. De plus, si $\boldsymbol{U} \in \mathcal{U}^{\mathrm{ps}}$, alors $\phi \in \Phi^{\mathrm{ps}}$. L'inclusion inverse s'obtient de la même manière. □

Au cours de ce manuscrit, nous userons régulièrement de l'interprétation fonctionnelle que nous venons d'expliciter. Celle-ci a en effet l'avantage de mettre en avant la dépendance entre les décisions et les observations. Le lien entre ces deux quantités sera appelé une stratégie, ou encore la décision sera dite en *feedback* sur la variable d'observation $\boldsymbol{Y}$. Nous comprenons mieux, à l'aide de cette interprétation, que nous sommes face à des problèmes d'optimisation en dimension infinie.

1.1.5 Problèmes à plusieurs pas de temps

On s'intéresse maintenant plus spécifiquement à des problèmes dynamiques sur un horizon de temps discret et fini $t_0, \dots, t_N = T$. Soit un système dynamique sur cet intervalle discret, caractérisé par l'équation d'évolution $\boldsymbol{X}_{t+1} = f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right)$, que l'on appellera aussi la dynamique du système. Par référence à l'application plus particulièrement traitée dans ce manuscrit, on appellera souvent $\boldsymbol{X}_t$ le stock, $\boldsymbol{U}_t$ la commande et $\boldsymbol{W}_t$ le bruit. Toutes trois sont des variables aléatoires définies sur un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$ et à valeurs dans des espaces de Hilbert $\mathbb{X}_t$, $\mathbb{U}_t$ et $\mathbb{W}_t$, et nous considérons toujours des commandes de carré intégrable. Partant d'un stock initial donné, on cherche à commander le système, à l'aide de $\boldsymbol{U}$, afin de minimiser un coût qui dépend de l'évolution du stock, de la commande et du bruit sur l'horizon de temps. Enfin, au cours de l'évolution du système, des observations sont faites sur celui-ci et la commande est autorisée à en dépendre. Il est alors naturel de chercher à utiliser cette information pour définir de meilleures décisions. Ainsi, à la différence de la boucle ouverte, nous parlerons maintenant de stratégies (qui à toute observation possible du système associent une décision), plutôt que de simples décisions.

Le problème à plusieurs pas de temps s'écrit :

$$\min_{\boldsymbol{X},\boldsymbol{U}} \quad \mathbb{E}\left(\sum_{t=0}^{T-1} C_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t\right) + K\left(\boldsymbol{X}_T\right)\right), \tag{1.3a}$$

$$\text{s.c.} \quad \boldsymbol{X}_{t+1} = f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right), \qquad \forall t = 0, \dots, T-1, \tag{1.3b}$$

$$\boldsymbol{X}_0 = \boldsymbol{W}_0, \tag{1.3c}$$

$$\boldsymbol{U}_t \text{ est } \sigma\left(h_t\left(\boldsymbol{U}, \boldsymbol{W}\right)\right)\text{-mesurable.} \tag{1.3d}$$

Les contraintes (1.3b) et (1.3c) sont $\mathbb{P}$-presque sûres. À ce modèle "simple" peuvent s'ajouter des contraintes supplémentaires sous différentes formes : en probabilité ou en espérance, que nous rencontrerons au chapitre 5, où d'autres contraintes presque-sûres, restreignant l'ensemble des décisions possibles à chaque pas de temps, comme ce sera le cas dans les chapitres 3 et 4.

Remarque 1.2 (Bruit et processus stochastiques non commandés). Bien que les équations dynamiques (1.3b) fassent apparaître tout à la fois le stock, la commande et le bruit, nous gardons à l'esprit qu'elles permettent de modéliser tant des processus commandés tels qu'un stock d'énergie que des processus non commandés tels qu'un prix de marché ou une demande en énergie sur lesquels nous n'aurions pas d'influence. Cependant, dans la suite de ce mémoire, nous nous concentrons plutôt sur la manière de produire des stratégies de commande que sur la modélisation des processus stochastiques.

Le processus de décision "naturel" nous impose de ne faire dépendre la décision à l'instant t que de l'information disponible à cet instant. C'est le principe de causalité : la décision à un certain instant ne peut dépendre que de réalisations d'aléas passés. Nous décrivons à nouveau les différents cas énoncés dans le cadre général pour ce problème dynamique.

Boucle ouverte On a par exemple $h_t(\boldsymbol{U}, \boldsymbol{W}) = 0$. La contrainte (1.3d) impose alors à $\boldsymbol{U}_t$ d'être une constante. Autrement dit, la décision n'a pas la possibilité de s'adapter aux aléas ; les décisions doivent être prises dès le tout premier instant.

Mémoire parfaite C'est le cas où $h_t(\boldsymbol{U}, \boldsymbol{W}) = (\boldsymbol{U}_0, \dots, \boldsymbol{U}_{t-1}, \boldsymbol{W}_0, \dots, \boldsymbol{W}_t)$. On aura donc :

$$\boldsymbol{U}_0 \text{ est } \sigma\left(\boldsymbol{W}_0\right)\text{-mesurable}, \dots, \boldsymbol{U}_t \text{ est } \sigma\left(\boldsymbol{W}_0, \dots, \boldsymbol{W}_t\right)\text{-mesurable.}$$

On est alors dans un cas en information statique, en boucle fermée. C'est le cas sur lequel nous travaillerons tout au long de ce manuscrit.

Oubli On peut connaître un cas intermédiaire où le décideur observe tout ou partie du système et oublie une partie de ses observations passées. Par exemple, on peut avoir $h_t(\boldsymbol{U}, \boldsymbol{W}) = (\boldsymbol{U}_{t-1}, \boldsymbol{W}_{t-1})$. Le décideur ne retient que sa dernière action et la valeur de l'aléa à l'instant précédent. Ce cas est donc en boucle fermée, comme le second cas, mais en information dynamique. Nous sommes de nouveau face aux problèmes esquissés au §1.1.2.

1.2 Résolution de problèmes de commande optimale stochastique

Nous présentons à présent quelques méthodes "classiques" permettant de traiter des problèmes d'optimisation stochastique dynamique tels que le problème (1.3). On parle encore de problèmes de commande optimale stochastique. On s'intéresse plus précisément à des problèmes en information statique et en mémoire parfaite. Au §1.2.1, nous présentons le principe de programmation dynamique qui sera utile tout au long de ce mémoire, et sur lequel nous nous attarderons de manière plus approfondie au chapitre 5. Puis, au §1.2.2, nous décrivons brièvement la méthodologie des arbres de scénarios et enfin, au §1.2.3, nous présentons les méthodes particulaires. Les deux dernières méthodes seront étudiées en détail au chapitre 2.

1.2.1 Cadre markovien : programmation dynamique

On suppose ici que le système est en mémoire parfaite. Notons $\mathcal{A}_t$ la tribu engendrée par le passé du bruit jusqu'à l'instant t : $\boldsymbol{W}_0, \dots, \boldsymbol{W}_t$, et notons $\mathcal{U}_t$ l'espace des variables aléatoires définies sur $(\Omega, \mathcal{A}, \mathbb{P})$, à valeurs dans $\mathbb{U}_t$, de carré intégrable, et mesurables par rapport à $\mathcal{A}_t$. Ainsi, le problème d'optimisation s'écrit :

$$\begin{aligned} \min_{\boldsymbol{X},\boldsymbol{U}} \quad & \mathbb{E}\left(\sum_{t=0}^{T-1} C_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t\right) + K\left(\boldsymbol{X}_T\right)\right), \\ \text{s.c.} \quad & \boldsymbol{X}_{t+1} = f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right), \qquad \forall t = 0, \dots, T-1, \\ & \boldsymbol{X}_0 = \boldsymbol{W}_0, \\ & \boldsymbol{U}_t \text{ est } \mathcal{A}_t\text{-mesurable.} \end{aligned}$$

On a alors que la commande $\boldsymbol{U}$ recherchée est, à chaque instant t, une fonction de tout le passé du bruit $\boldsymbol{W}_0, \dots, \boldsymbol{W}_t$. Sans même parler d'optimisation, rien que l'évaluation d'une telle quantité paraît difficile en pratique. Le principe de programmation dynamique est un premier pas permettant de réduire, dans certains cas, la dimension de l'espace de départ associé à la stratégie.

On appelle fonction valeur ou fonction de Bellman à l'instant t, pour tout t allant de t_0 à T, la fonction $V_t : \mathbb{X}_t \to \mathbb{R}$ définie par :

$$V_t(x) := \min_{\boldsymbol{U} \in \mathcal{U}_t^{T-1}} \mathbb{E}\left(\sum_{s=t}^{T-1} C_s\left(\boldsymbol{X}_s, \boldsymbol{U}_s\right) + K\left(\boldsymbol{X}_T\right) \,\middle|\, \boldsymbol{X}_t = x\right), \qquad \forall x \in \mathbb{X}, \tag{1.4}$$

avec $\mathcal{U}_t^{T-1} = \mathcal{U}_t \times \cdots \times \mathcal{U}_{T-1}$ et sous les contraintes de dynamique (1.3b). Cette fonction représente donc le coût optimal du problème, partant à l'instant t avec un stock x. Le

principe de programmation dynamique donne une relation de récurrence arrière liant les fonctions de valeur aux différents pas de temps.

Théorème 1.4 (Principe de Programmation Dynamique)**.** *Supposons que les bruits sont indépendants pas de temps par pas de temps. Alors la stratégie optimale à l'instant t ne dépend du passé du bruit qu'à travers la variable $\boldsymbol{X}_t$, alors appelée variable d'état du système. De plus, on a l'équation de la programmation dynamique suivante.*

$$V_T(x) = K(x), \qquad \forall x \in \mathbb{X}_t, \tag{1.5a}$$

$$V_t(x) = \mathbb{E}\left(\min_{u \in \mathbb{U}_t} C_t(x,u) + V_{t+1}(f_t(x,u,\boldsymbol{W}_{t+1}))\right), \qquad \forall x \in \mathbb{X}_t, \forall t = t_0, \dots, T-1. \tag{1.5b}$$

Ainsi, sous l'hypothèse d'indépendance des bruits (voir Remarque 1.3), le principe de programmation dynamique nous indique que la variable $\boldsymbol{X}_t$ est, à l'instant t, une statistique suffisante pour calculer la stratégie optimale à l'instant t du problème (1.3). Ce principe a d'abord été énoncé par Bellman (1957) ; il est central en contrôle optimal et on trouve un grand nombre d'excellents ouvrages à ce sujet, dont ceux de Bertsekas (2000), Whittle (1982), Puterman (1994). On pourra également consulter les cours de Quadrat et Viot (1999), Quadrat (2007).

Remarque 1.3 (Blanchiment du bruit). Il n'y a pas a priori de raison pour laquelle les variables aléatoires $\boldsymbol{W}_0, \dots, \boldsymbol{W}_T$ seraient indépendantes pas de temps par pas de temps. Cependant, il est classique en commande optimale stochastique d'avoir recours à un processus dit de blanchiment [4] du bruit, qui revient à inclure dans la variable $\boldsymbol{X}_t$ l'information suffisante pour que les bruits soient indépendants en temps. Par exemple, supposons que le bruit soit un processus réel tel que $\boldsymbol{W}_{t+1} = \alpha \boldsymbol{W}_t + \boldsymbol{\varepsilon}_{t+1}$ avec $\boldsymbol{\varepsilon}_{t+1}$ indépendant de $\boldsymbol{W}_0, \dots, \boldsymbol{W}_t$, et ce pour tout $t = 0, \dots, T-1$. Dans ce cas, on posera comme "nouvelle variable de stock" : $\widetilde{\boldsymbol{X}}_t = (\boldsymbol{X}_t, \boldsymbol{W}_t)$ et comme "nouveau bruit" $\boldsymbol{\varepsilon}_t$, de sorte que l'on aura la nouvelle dynamique de "stock" :

$$\widetilde{\boldsymbol{X}}_{t+1} := \begin{pmatrix} \boldsymbol{X}_{t+1} \\ \\ \boldsymbol{W}_{t+1} \end{pmatrix} = \begin{pmatrix} f_t(\boldsymbol{X}_t, \boldsymbol{U}_t, \alpha \boldsymbol{W}_t + \boldsymbol{\varepsilon}_{t+1}) \\ \\ \alpha \boldsymbol{W}_t + \boldsymbol{\varepsilon}_{t+1} \end{pmatrix}.$$

Les variables aléatoires $\boldsymbol{W}_0, \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_T$ qui constituent le "nouveau" processus de bruit sont maintenant indépendantes en temps. Cette procédure de blanchiment fait qu'au plus, si $\boldsymbol{W}_{t+1}$ dépend de tout la passé du bruit, on devra inclure tout le passé du bruit dans la variable $\boldsymbol{X}$. Ainsi, quitte à devoir préalablement blanchir le processus de bruit, on peut supposer l'indépendance en temps des bruits.

L'équation de la programmation dynamique nous donne un moyen de calculer les fonctions valeur ainsi que les stratégies optimales de manière rétrograde. Remarquons qu'en parallèle de la résolution de cette équation, on obtient la stratégie optimale comme une fonction de $\boldsymbol{X}_t$ à l'instant t. Il s'agit d'un grand pas en avant en terme de complexité par rapport au problème initial car il nous indique qu'on peut rechercher la stratégie optimale comme une fonction de la variable $\boldsymbol{X}_t$ et non plus de tout le passé du bruit. Or,

4. Le terme *blanchiment* est quelque peu abusif au sens où il fait référence à la notion de processus de bruit blanc en statistique. Or ceux-ci sont non seulement indépendants en temps, mais sont de plus centrés réduits.

la dimension de l'espace $\mathbb{X}_t$ est en général constante en temps. On appelle maintenant $\boldsymbol{X}_t$ la variable d'état du système : elle résume l'information nécessaire à la prise de décision optimale.

Cependant, la résolution de cette équation ne peut en général pas se faire de manière analytique et il est nécessaire de trouver un moyen de représenter les fonctions valeur sur un ordinateur. On a typiquement recours à une discrétisation de l'espace $\mathbb{X}_t$. On représente alors la fonction de Bellman par sa valeur en chaque point d'une grille suffisamment fine sur l'espace $\mathbb{X}_t$. Or, la complexité d'une telle procédure dépend clairement de manière exponentielle de la dimension de l'espace d'état. Cette propriété est connue sous le nom de *malédiction de la dimension* (*curse of dimensionality* en anglais). Bien qu'il soit difficile de donner une barrière absolue, il est généralement impossible de traiter numériquement des problèmes dont la dimension de l'état dépasse 5, environ. Citons tout de même les récents résultats de Vezolle, Vialle, et Warin (2009) qui permettent de repousser quelque peu cette barrière, en tirant parti du calcul parallèle.

Plusieurs développements récents proposent des approximations de la fonction valeur permettant de faire face, dans certains cas, à la malédiction de la dimension. Parmi elles on peut citer la programmation dynamique approximée, qui est une idée originale de Bellman et Dreyfus (1959), et qui consiste à rechercher les fonctions valeur à chaque instant comme des combinaisons linéaires de fonctions de base choisies à l'avance. On évite ainsi le calcul et le stockage de la fonction sur une grille. Un grand nombre de travaux sont consacrés à cette méthode (voir, parmi d'autres, de Farias et Van Roy, 2003, Longstaff et Schwartz, 2001, Tsitsiklis et Van Roy, 1999). On pourra également consulter à ce sujet les ouvrages de Bertsekas et Tsitsiklis (1996) ou de Powell (2007). L'inconvénient de ce type de méthodes reste qu'il faut choisir a priori une base de fonctions avec laquelle approcher la fonction valeur. Or, nous n'avons généralement que peu d'idées sur la forme de celle-ci.

Il faut également noter que l'équation de programmation dynamique ne peut être directement mariée aux techniques de décomposition de grands systèmes bien connues dans le cadre déterministe (voir le cours de Cohen, 2004, et le §3.1). En effet, en utilisant de telles techniques, on est capable de décomposer l'espace des commandes $\mathbb{U}_t$, qui est l'espace d'arrivée des stratégies que nous recherchons. Mais la complexité liée à la résolution de l'équation de programmation dynamique provient plutôt de la dimension de l'espace de départ, qui est l'espace d'état $\mathbb{X}_t$. Or, la dimension de ce dernier n'est pas diminuée à l'issue de l'application d'une technique de décomposition.

Dans le chapitre 3, nous proposerons une manière de lier programmation dynamique et décomposition qui peut, parce qu'elle contraint la variable de décomposition à appartenir à une certaine classe a priori, rappeler la programmation dynamique approximée. C'est d'ailleurs la raison pour laquelle nous avons choisi, pour l'algorithme en question, le nom de *Dual Approximate Dynamic Programming*.

1.2.2 Arbres de scénarios

Comme le montre Barty (2004) dans sa thèse, lorsque l'on souhaite discrétiser un problème d'optimisation stochastique dynamique, il faut bien comprendre que l'on s'attaque à deux objets probabilistes de natures différentes, qui peuvent éventuellement être traités séparément :

1. l'espérance présente dans le critère, qui est généralement estimée par Monte-Carlo ;
2. les contraintes de mesurabilité, qui sont elles plus délicates à traiter.

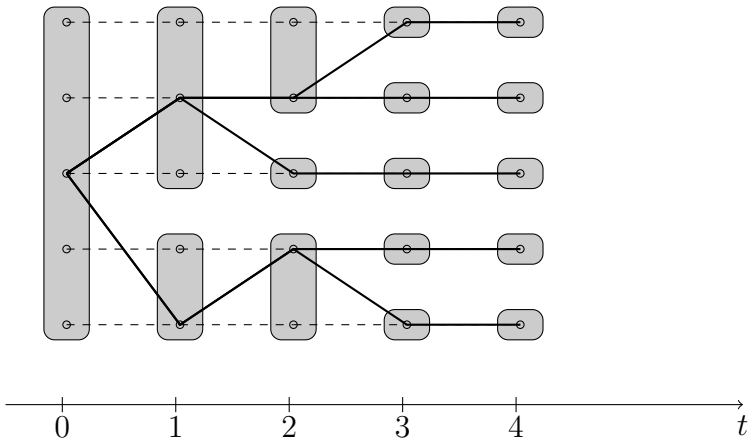

FIGURE 1.1 – Construction d'un arbre de scénarios

L'étude de la discrétisation de ces objets, et notamment de la contrainte de mesurabilité, a fait l'objet de plusieurs articles, dont ceux de Pennanen (2005), Carpentier, Chancelier, et De Lara (2009a), Heitsch, Römisch, et Strugarek (2006). Dans le cas de plusieurs pas de temps, en mémoire parfaite, la méthodologie "arbres de scénarios", sur laquelle s'appuie la communauté *Stochastic Programming*, propose de discrétiser ces deux objets à la fois en utilisant des chroniques organisées en scénarios arborescents. Ceci n'est possible qu'en information statique. Nous présentons informellement leur mise en œuvre en nous appuyant sur le schéma de la figure 1.1.

On se donne un ensemble d'échantillons de la variable aléatoire $\boldsymbol{W}$, c'est-à-dire formellement un ensemble de variables aléatoires $\boldsymbol{W}^1, \dots, \boldsymbol{W}^N$ i.i.d. de même loi que $\boldsymbol{W}$. Dans la figure 1.1, nous avons choisi $N = 5$ et un horizon de 5 pas de temps, les processus étant symbolisés par les petits cercles reliés par des pointillés. Le fait de choisir un échantillon de taille N correspond à une estimation de type Monte-Carlo. Afin de représenter la contrainte de non-anticipitivité (1.3d), on regroupe ces scénarios sur la base de leur passé commun, en utilisant une distance ad hoc (voir Heitsch *et al.*, 2006). Les rectangles gris constituent ces classes. Puis, on choisit un représentant parmi les différents échantillons de bruit à cet instant. Nous avons maintenant une structure arborescente qui est représentée par les traits gras, le représentant de chaque classe étant le point par lequel passe ce trait. À chaque nœud de l'arbre correspond un passé unique ; on définit une variable de stock et une variable de commande en chaque nœud. Puis on réécrit l'ensemble des contraintes du problème (1.3) sur cet arbre, ainsi que la fonction objectif[5]. Le problème ainsi posé, bien "qu'indexé" par le tirage d'échantillons du bruit effectué, est maintenant un problème de programmation mathématique "classique". On le résout alors par une méthode ad hoc et on obtient des valeurs optimales pour la commande et pour le stock en chaque nœud, ainsi que la valeur du problème sur l'arbre qui se veut être une estimation de la valeur du problème de départ.

Cette méthodologie a un certain nombre d'avantages, parmi lesquels la simplicité de sa mise en œuvre. De plus, une fois l'arbre construit, on a à notre disposition tous les outils de l'optimisation mathématique "classique" (programmation linéaire, techniques variationnelles) pour résoudre le problème discrétisé. On peut, en particulier, faire appel aux techniques dé décomposition (voir le §3.1.3 pour un état de l'art).

5. À ce stade, on peut bien sûr définir des poids probabilistes aux arêtes.

Mais on entrevoit aussi l'une des difficultés d'une telle méthodologie dans le cadre de la commande optimale : elle ne fournit pas (directement) de solution sous forme de stratégie. On peut bien sûr chercher à interpoler les valeurs de commande obtenues en chaque nœud, mais cela amène, comme nous le montrerons dans le chapitre 2, à des contrôles de mauvaise qualité. Nous revenons, lors de ce chapitre, plus en détails sur la mise en œuvre des arbres de scénarios.

Enfin, il faut garder à l'esprit que les arbres de scénarios constituent une méthode stochastique de résolution, en ce sens qu'ils fournissent une solution qui dépend du tirage de scénarios effectué pour construire l'arbre. Cela rend l'évaluation de la qualité de la solution plus délicat (il faut souvent faire un grand nombre d'expériences indépendantes pour évaluer l'erreur).

Pour plus de précisions, notamment concernant les résultats de convergence de telles méthodes d'échantillonnage, on renvoie le lecteur vers l'ouvrage de Birge et Louveaux (1997) ou le recueil de Shapiro, Dentcheva, et Ruszczyński (2009) pour une vision générale de la programmation stochastique, ou encore vers les travaux de Pflug (2001), Heitsch et Römisch (2003), Dupačová, Gröwe-Kuska, et Römisch (2003) pour la construction et la réduction des arbres de scénarios.

1.2.3 Méthodes particulaires

Nous présentons maintenant une autre manière de considérer la résolution de problèmes de commande optimale stochastique. Suivant une approche de type variationnel, nous décrivons des conditions d'optimalité d'ordre 1, c'est-à-dire faisant intervenir le gradient de la fonction objectif et des contraintes. La présentation faite ici s'appuie sur les travaux de Barty (2004) et de Dallagi (2007).

Conditions d'optimalité

Oublions un instant la structure dynamique du problème pour considérer à nouveau le problème (1.1). Supposons qu'un aléa survienne et soit observé (éventuellement en partie seulement). Au regard de cette observation, une décision est prise. Un coût est alors infligé au système, dépendant des valeurs de l'aléa et de la décision prise. La commande peut être généralement soumise à deux types de contraintes : des contraintes "ponctuelles" et des contraintes d'information. Nous rappelons le problème (1.1).

$$\begin{aligned} \min_{\boldsymbol{U}\in\mathcal{U}} \quad & \mathbb{E}\left(j\left(\boldsymbol{U},\boldsymbol{W}\right)\right) \\ \text{s.c.} \quad & \boldsymbol{U}\in\mathcal{U}^{\text{ps}}\cap\mathcal{U}^{\text{mes}}. \end{aligned}$$

Nous cherchons des conditions d'optimalité pour le problème (1.1). Nous rappelons ici le cheminement amenant à celles-ci, dont on trouvera une étude plus détaillée dans la thèse de Dallagi (2007, Chapitre II) ou bien dans l'article de Carpentier, Cohen, et Dallagi (2009b). Pour énoncer des conditions d'optimalité, il va falloir nous intéresser à la projection sur l'ensemble $\mathcal{U}^{\text{ps}}\cap\mathcal{U}^{\text{mes}}$. Un premier lemme nous indique que nous pouvons effectuer la projection de la commande sur l'ensemble des contraintes ponctuelles "ω par ω".

Lemme 1.5 (Dallagi, 2007, Lemme II.5)**.** *Soit* $\boldsymbol{\Gamma} : \Omega \rightrightarrows \mathbb{U}$ *une multi-application mesurable, à valeurs convexes fermées. Le sous-ensemble* $\mathcal{U}^{ps}\subset\mathcal{U}$ *défini par :*

$$\mathcal{U}^{ps} := \left\{\boldsymbol{U}\in\mathcal{U}, \boldsymbol{U}\in\boldsymbol{\Gamma}, \mathbb{P}\text{-}p.s.\right\},$$

est un convexe fermé de $\mathcal{U}$ *et on a que :*

$$\left(\Pi_{\mathcal{U}^{ps}}\left(\boldsymbol{U}\right)\right)(\omega) = \Pi_{\boldsymbol{\Gamma}(\omega)}\left(\boldsymbol{U}\left(\omega\right)\right), \qquad \mathbb{P}\text{-}p.s.$$

Le résultat que nous présentons maintenant s'intéresse à la manière de projeter sur l'intersection des contraintes ponctuelles et des contraintes de mesurabilité.

Lemme 1.6 (Dallagi, 2007, Lemme A.6)**.** *Soit* $\mathcal{U}^{ps}$ *un convexe fermé de* $\mathcal{U}$ *et* $\mathcal{U}^{mes}$ *un sous-espace fermé de* $\mathcal{U}$ *tels que* $\Pi_{\mathcal{U}^{ps}}(\mathcal{U}^{mes}) \subset \mathcal{U}^{mes}$*. Alors :*

$$\Pi_{\mathcal{U}^{ps}\cap\mathcal{U}^{mes}} = \Pi_{\mathcal{U}^{ps}} \circ \Pi_{\mathcal{U}^{mes}}.$$

À l'aide de ces outils, on peut énoncer une condition nécessaire d'optimalité d'ordre 1 pour le problème (1.1). Nous faisons référence à l'article de Hiriart-Urruty (1982) pour les conditions d'optimalité associées à des problèmes de minimisation d'une fonction intégrable dans un cadre non-différentiable.

Proposition 1.7 (Dallagi, 2007, Proposition II.4)**.** *Supposons que :*

1. *la multi-application* $\boldsymbol{\Gamma} : \Omega \rightrightarrows \mathbb{U}$*, qui permet de définir l'ensemble* $\mathcal{U}^{ps}$ *des contraintes ponctuelles, est* $\mathcal{G}$*-mesurable à valeurs convexes fermées ;*
2. $\mathcal{G} \subset \mathcal{A}$ *;*
3. $j(\cdot, \boldsymbol{W})$ *est* $\mathcal{C}^1(\mathbb{U})$ $\mathbb{P}$*-p.s. ;*
4. j *est s.c.i. sur* $\mathbb{U} \times \mathbb{W}$ *;*
5. $j'_u(\boldsymbol{U}, \boldsymbol{W}) \in \mathcal{U}$*,* $\forall \boldsymbol{U} \in \mathcal{U}$*.*

Si $\boldsymbol{U}^*$ *est solution de* (1.1)*, alors :*

$$\mathbb{E}\left(j'\left(\boldsymbol{U}^*, \boldsymbol{W}\right) \mid \boldsymbol{Y}\right) \in -\partial\chi_{\mathcal{U}^{ps}}\left(\boldsymbol{U}^*\right).$$

On peut réécrire cette condition d'optimalité à l'aide d'une projection sur l'ensemble admissible $\mathcal{U}^{\text{ps}}$:

$$\exists \varepsilon > 0, \boldsymbol{U} = \Pi_{\mathcal{U}^{\text{ps}}}\left(\boldsymbol{U} - \varepsilon\mathbb{E}\left(\nabla J\left(\boldsymbol{U}\right) \mid \boldsymbol{Y}\right)\right),$$

qui peut se faire, à l'aide du lemme 1.5, "ω par ω". L'idée des méthodes particulaires est de déduire de ces conditions un algorithme de gradient.

Nous spécifions maintenant cette condition d'optimalité au cas dynamique qui nous intéresse dans cette étude. Soit, pour tout $t = 0, \ldots, T-1$, $\mathcal{A}_t := \sigma\left(\boldsymbol{W}_0, \ldots, \boldsymbol{W}_t\right)$. On rappelle le problème de commande optimale stochastique (1.3) qui nous intéresse[6] :

$$\begin{aligned}
\min_{\boldsymbol{X}\in\mathcal{X},\boldsymbol{U}\in\mathcal{U}} \quad & \mathbb{E}\left(\sum_{t=0}^{T-1} C_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t\right) + K\left(\boldsymbol{X}_T\right)\right) \\
\text{s.c.} \quad & \boldsymbol{X}_0 = \boldsymbol{W}_0, \\
& \boldsymbol{X}_{t+1} = f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right), \qquad \forall t = 0, \ldots, T-1, \\
& \boldsymbol{U}_t \preceq \mathcal{A}_t, \qquad \forall t = 0, \ldots, T-1, \\
& \boldsymbol{U}_t \in \mathcal{U}_t^{\text{ps}}, \qquad \forall t = 0, \ldots, T-1.
\end{aligned}$$

Si on spécifie la proposition 1.7 au problème dynamique, on obtient le résultat suivant :

Proposition 1.8. *Si :*

6. On considère dorénavant que l'on est en mémoire parfaite.

– $\mathcal{U}_t^{ps}$ *est un sous-ensemble convexe fermé de* $\mathcal{U}_t$ *;*
– *les coûts et les dynamiques du problème sont de classe* $\mathcal{C}^1$ *et de carré intégrable ;*
– $\mathbf{\Gamma}_t^{ps}$ *est* $\mathcal{A}_t$*-mesurable, pour tout* $t = 0, \dots, T-1$*,*

Alors il existe un processus aléatoire $\mathbf{\Lambda}$*, où* $\mathbf{\Lambda}_t$ *est élément de* $L^2(\Omega, \mathcal{A}, \mathbb{P}\,; \mathbb{X}_t)$*, tel que :*

$$\boldsymbol{X}_0 = \boldsymbol{W}_0, \tag{1.6a}$$
$$\boldsymbol{X}_{t+1} = f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right), \tag{1.6b}$$
$$\mathbf{\Lambda}_T = \nabla K\left(\boldsymbol{X}_T\right), \tag{1.6c}$$
$$\mathbf{\Lambda}_t = \frac{\partial C_t}{\partial x}\left(\boldsymbol{X}_t, \boldsymbol{U}_t\right)^\top + \mathbb{E}\left(\frac{\partial f_t}{\partial x}\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right)^\top \mathbf{\Lambda}_{t+1} \,\middle|\, \mathcal{A}_t\right), \tag{1.6d}$$
$$\frac{\partial C_t}{\partial u}\left(\boldsymbol{X}_t, \boldsymbol{U}_t\right)^\top + \mathbb{E}\left(\frac{\partial f_t}{\partial u}\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right)^\top \mathbf{\Lambda}_{t+1} \,\middle|\, \mathcal{A}_t\right) \in -\partial \chi_{\mathcal{U}_t^{ps}}\left(\boldsymbol{U}_t\right). \tag{1.6e}$$

La démonstration est donnée par Dallagi (2007, Théorème II.16). Le processus aléatoire $\mathbf{\Lambda}$, qui est de la même dimension que l'état, est appelé l'état adjoint.

L'état adjoint représente la sensibilité du coût optimal à une variation infinitésimale de l'état. À l'aide de cette interprétation économique on comprend bien l'équation (1.6c), puisqu'arrivé au dernier pas de temps, une variation infinitésimale de l'état de système n'aurait d'autre effet que celui de faire varier le coût au dernier pas de temps. À l'image de l'équation de programmation dynamique[7], on a naturellement une relation rétrograde en temps liant l'état adjoint à un certain instant à son successeur : l'influence d'une variation infinitésimale de l'état à l'instant t se ressent à la fois à travers le coût à l'instant t mais également à travers les coûts futurs, du fait de la dynamique de l'état (1.3b).

On peut obtenir à partir de (1.6) des conditions spécifiques au cadre markovien. On suppose, comme pour le principe de programmation dynamique, que les bruits sont indépendants pas de temps par pas de temps. On peut alors, sous des hypothèses essentiellement similaires à celles de la proposition 1.8 (voir Carpentier *et al.*, 2009b, Théorème 2.6), réécrire les conditions d'optimalité (1.6) de la façon suivante. On a les dynamiques progrades :

$$\boldsymbol{X}_0 = \boldsymbol{W}_0, \tag{1.7a}$$
$$\boldsymbol{X}_{t+1} = f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right), \tag{1.7b}$$

les dynamiques rétrogrades :

$$\mathbf{\Lambda}_T = \frac{\partial C_T}{\partial x}\left(\boldsymbol{X}_T, \boldsymbol{U}_T\right)^\top, \tag{1.7c}$$
$$\mathbf{\Lambda}_t = \frac{\partial C_t}{\partial x}\left(\boldsymbol{X}_t, \boldsymbol{U}_t\right)^\top + \mathbb{E}\left(\frac{\partial f_t}{\partial x}\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right)^\top \mathbf{\Lambda}_{t+1} \,\middle|\, \boldsymbol{X}_t\right), \tag{1.7d}$$

et la condition sur le gradient :

$$\frac{\partial C_t}{\partial u}\left(\boldsymbol{X}_t, \boldsymbol{U}_t\right)^\top + \mathbb{E}\left(\frac{\partial f_t}{\partial u}\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right)^\top \mathbf{\Lambda}_{t+1} \,\middle|\, \boldsymbol{X}_t\right) \in -\partial \delta_{\mathcal{U}_t^{\mathrm{ps}}}\left(\boldsymbol{U}_t\right). \tag{1.7e}$$

7. Il existe bien sûr, sous certaines hypothèses, un lien étroit entre l'équation de la programmation dynamique et les conditions d'optimalité (1.6). Ce lien est très clairement mis en évidence par Bertsekas (2000).

On a donc qu'à l'optimum, conditionner par rapport à l'état optimal $\boldsymbol{X}_t$, tel que c'est le cas dans les équations (1.7d) et (1.7e), revient à conditionner par rapport au passé du bruit. Il faut cependant être attentif au fait que le gradient mis en évidence dans la relation (1.7e) n'est un gradient qu'à l'optimum.

Méthodes particulaires

Il existe plusieurs versions de méthodes particulaires, qui correspondent aux différentes versions des conditions d'optimalité (nous venons d'en citer deux). Elles diffèrent dans leur mise en œuvre ainsi que dans leur complexité, mais consistent toutes en une discrétisation de conditions d'optimalité, à l'aide de scénarios que l'on appelle ici particules. Nous introduisons la version qui nous servira au cours du chapitre 2, que l'on appelle version markovienne adaptée, ou encore "espérance du gradient, espérance de l'état adjoint".

On se donne N variables aléatoires $\boldsymbol{W}^1, \dots, \boldsymbol{W}^N$ i.i.d. de même loi que $\boldsymbol{W}$, que nous appelons échantillons de $\boldsymbol{W}$. La méthode particulaire est une méthode itérative qui calcule de manière prograde des particules d'état, pour tout $i = 1, \dots, N$:

$$\boldsymbol{X}_{t+1}^i = f_t\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^i\right), \tag{1.8a}$$

$$\boldsymbol{X}_0^i = \boldsymbol{W}_0^i, \tag{1.8b}$$

de manière rétrograde des particules d'état adjoint :

$$\boldsymbol{\Lambda}_T^i = K'\left(\boldsymbol{X}_T^i\right), \tag{1.8c}$$

$$\boldsymbol{\Lambda}_t^i = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{\partial C_t}{\partial x}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i\right) + \left(\tilde{\boldsymbol{\Lambda}}_{t+1}^{i,j}\right)^\top \frac{\partial f_t}{\partial x}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j\right)\right), \tag{1.8d}$$

et cherche à vérifier, à l'issue du processus itératif, la condition sur le gradient :

$$\sum_{j=1}^{N}\left(\frac{\partial C_t}{\partial u}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i\right) + \left(\tilde{\boldsymbol{\Lambda}}_{t+1}^{i,j}\right)^\top \frac{\partial f_t}{\partial u}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j\right)\right) \in -\partial\chi_{\mathcal{U}_t^{\mathrm{ps}}}\left(\boldsymbol{U}_t^i\right), \tag{1.8e}$$

où $\tilde{\boldsymbol{\Lambda}}_{t+1}^{i,j}$ est une estimation de la variable aléatoire $\mathbb{E}\left(\boldsymbol{\Lambda}_{t+1} | \, \boldsymbol{X}_t = f_t(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j)\right)$ à l'aide des échantillons de la variable d'état $(\boldsymbol{X}_{t+1}^i)_i$ et de l'état adjoint $(\boldsymbol{\Lambda}_{t+1}^i)_i$.

La procédure est la suivante. Supposons que l'on ait des particules de commande $\boldsymbol{U}_t^i$.

1. On propage l'état en utilisant les dynamiques (1.8a) et les particules de commande courantes.
2. On propage l'état adjoint en utilisant les dynamiques (1.8d), les particules d'état que nous venons de calculer, et les particules de commande courantes.
3. On calcule les particules de gardient :

$$\boldsymbol{G}_t^i = \sum_{j=1}^{N}\left(\frac{\partial C_t}{\partial u}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i\right) + \left(\tilde{\boldsymbol{\Lambda}}_{t+1}^{i,j}\right)^\top \frac{\partial f_t}{\partial u}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j\right)\right).$$

4. On met à jour les particules de commande par la règle :

$$\left(\boldsymbol{U}_t^i\right)^+ = \Pi_{\mathcal{U}_t^{\mathrm{ps}}}\left[\boldsymbol{U}_t^i - \rho \times \boldsymbol{G}_t^i\right],$$

avec ρ un paramètre réel.

Ainsi, si la méthode particulaire converge, on a trouvé une stratégie qui vérifie les conditions (1.8). Au cours de cette procédure, on a à chaque pas de temps N particules d'état et de commande qui permettent, à l'aide d'opérateurs de régression, de synthétiser un contrôle sous la forme d'un *feedback* sur l'état $\boldsymbol{X}$. Contrairement à la programmation dynamique, on n'opère des calculs que pour un nombre N fixé de valeurs possibles d'état. Or, pour certains types de problèmes, même en grande dimension, il y a lieu de penser que l'état optimal se concentre dans certaines régions de l'espace. Nous observerons par exemple cette propriété dans les exemples numériques des chapitres 2 et 4.

Nous reviendrons en détail sur la mise en œuvre de cette méthode au §2.3, en particulier sur la manière dont sont effectuées les opérations de régression sur l'état adjoint.

1.3 Organisation du mémoire

On présente maintenant le plan du reste du mémoire. Au cours du chapitre 2, nous nous intéressons à la comparaison de deux méthodes numériques basées sur l'usage de scénarios pour discrétiser les problèmes d'optimisation stochastique dynamique : les arbres de scénarios et les méthodes particulaires. Après avoir défini une notion d'erreur pour des lois de commande, nous montrons que les stratégies issues des méthodes particulaires "souffrent" moins d'un accroissement de l'horizon de temps que les arbres de scénarios. Ce constat rend les premières plus adaptées que les secondes aux problèmes dynamiques.

Partant du constat que, même avec les méthodes particulaires, nous faisons toujours face à la *malédiction de la dimension* inhérente aux problème de commande optimale[8], nous adaptons dans le chapitre 3 des résultats bien connus dans le cas déterministe, concernant la décomposition des grands systèmes, au cas de la boucle fermée. Cette application n'est pas directe et nécessite d'approcher la variable duale qui est, dans le cadre de notre étude, un processus stochastique. Nous proposons un algorithme qui permet de construire des lois de commande pour des problèmes de grande taille, et étudions sa convergence. Nous traitons ensuite, au chapitre 4, un tel problème à l'aide de l'algorithme proposé. Ce problème de gestion dynamique de portefeuille consiste à placer la production d'un grand nombre de réserves d'énergie à un horizon pluri-annuel, tout en garantissant l'équilibre production-demande du système à chaque instant, le tout devant se faire au moindre coût.

Enfin, au chapitre 5, nous étudions une propriété structurelle des problèmes d'optimisation dynamique : la consistance dynamique. Après avoir défini de façon informelle cette notion, qui n'est pas nouvelle mais qui a suscité beaucoup d'intérêt dans la littérature récente, nous dressons un parallèle avec le concept de structure d'état, concept central en commande optimale. Cela nous permet de montrer, pour une classe assez importante de problèmes d'optimisation stochastique dynamique, que l'on peut toujours bénéficier de cette propriété, quitte à changer la variable d'état du système.

Remarque 1.4. Un travail dont nous ne parlerons pas ici a pour autant fait l'objet d'un article publié pendant la durée de la thèse dans la revue *Monte Carlo Methods and Applications* (Barty, Girardeau, Roy, et Strugarek, 2008). Il s'agit d'une application de la méthode du gradient stochastique en boucle fermée proposée par Barty, Roy, et Strugarek (2007) à un problème de mathématiques financières : la valorisation d'options américaines. Cela avait fait l'objet d'une partie de mon stage de césure, encadré par Kengy Barty, Jean-Sébastien Roy et Cyrille Strugarek, au sein du département OSIRIS d'EDF R&D, alors

8. même si elle ne se présente pas tout à fait de la même façon dans le cas des méthodes particulaires que dans le cas de la programmation dynamique

que j'étais étudiant ingénieur à l'ENSTA.

Chapitre 2

Vitesse de convergence des méthodes à base de scénarios

> The record of a month's roulette playing at Monte Carlo can afford us material for discussing the foundations of knowledge.
>
> Karl Pearson (1857-1936)

Ce chapitre reprend l'essentiel d'un article (Girardeau, 2010) publié sur le site SPePS et soumis à *Optimization and Engineering* en février 2010.

Nous nous intéressons dans cette partie à la résolution numérique de problèmes de commande optimale stochastique à l'aide de méthodes basées sur des tirages des aléas du problème, et aux vitesses de convergence associées. À l'image de l'algorithme de Monte-Carlo pour le calcul d'espérance, on cherche à approcher la structure aléatoire du problème à l'aide d'un ensemble de réalisations des aléas tirés a priori, avant l'étape de résolution à proprement parler. Nous qualifions de tels procédés de méthodes stochastiques de résolution, qui sont à opposer aux méthodes déterministes telles que la programmation dynamique. Il faut être conscient que la solution apportée par une méthode stochastique dépend du tirage d'aléas effectué a priori, et cette solution est donc une variable aléatoire. Ainsi, il apparaît naturellement, lors de l'évaluation de la performance d'une telle méthode, un terme de type "variance" représentant la sensibilité de la solution vis-à-vis des tirages d'aléas. Cette variance s'ajoute au terme classique de biais de la stratégie solution, que l'on observe également pour une méthode déterministe.

Pour un problème de commande optimale stochastique tel que nous le définissons au §1.1.5, il est assez courant de chercher à représenter la diffusion des "futurs possibles" des aléas sous la forme d'un arbre de scénarios, comme nous l'avons déjà rapidement présenté au §1.2.2. On procède ensuite par une méthode ad hoc à la résolution du problème sur l'arbre. De telles modélisations remontent à Dantzig (1955) et ont été largement utilisées par la suite ; elles sont souvent rassemblées sous le terme de programmation stochastique (*Stochastic Programming* en anglais). On pourra consulter le très complet recueil de Ruszczyński et Shapiro (2003) pour de plus amples détails à ce sujet. La question de la convergence de tels schémas de discrétisation vers le problème original a été abordée par Pennanen (2005) ainsi que dans la thèse de Barty (2004). Dans cette dernière, on trouve également l'écriture de conditions d'optimalité pour ces problèmes, qui nous seront utiles dans ce chapitre.

La méthodologie des arbres de scénarios est une approche du type "discrétisation-optimisation" : on discrétise d'abord la structure aléatoire du problème à l'aide de scénarios. Puis, on transcrit les contraintes et la fonction objectif du problème original sur la structure discrétisée et on résout ce problème, où les variables aléatoires du problème original ont été échantillonnées. Autrement dit, on résout le problème pour une réalisation particulière des aléas, et le résultat est donc une réalisation de la solution. Les arbres de scénarios constituent plus une méthodologie qu'un algorithme, car on ne précise pas en général la manière dont sera résolu le problème discrétisé. L'intérêt principal d'une méthodologie telle que les arbres de scénarios est qu'elle permet de se ramener à un problème déterministe, que l'on peut résoudre par les méthodes classiques. On peut par exemple appliquer les méthodes de décomposition-coordination existant dans le cas déterministe (voir, parmi d'autres, Carpentier, Cohen, et Culioli, 1995, Higle et Sen, 1996, Bacaud, Lemaréchal, Renaud, et Sagastizábal, 2001, Emiel et Sagastizábal, 2010) pour tenter de résoudre des problèmes de grande taille.

Cette méthodologie connait un certain nombre d'avantages. Il n'est pas requis que les bruits soient indépendants pas de temps par pas de temps, contrairement à la programmation dynamique, et donc il n'est pas nécessaire de "blanchir" le bruit en augmentant la taille de l'état. De plus, la simplicité de sa mise en œuvre a certainement favorisé sa large utilisation dans l'industrie [1]. Cependant, on constate en pratique que les méthodes basées sur les arbres de scénarios nécessitent, pour atteindre une précision donnée, un nombre de scénarios (pour construire l'arbre) croissant de manière exponentielle avec l'horizon de temps. L'un des objets de ce chapitre est de quantifier et de mettre en lumière ce constat pratique. Nous travaillerons ici sur l'erreur au sens de la distance entre la stratégie optimale et la stratégie approchée par méthode d'arbre de scénarios ; Shapiro (2006) obtient des conclusions analogues aux nôtres en s'intéressant à la différence de performance sur la fonction objectif, via un résultat de grandes déviations.

Par la suite, nous montrons que cette propriété n'est pas liée au problème original mais à la manière dont il est discrétisé. Ainsi, si nous opérons la même étude d'erreur sur les méthodes particulaires introduites dans la thèse de Dallagi (2007), nous observons que le nombre de scénarios nécessaires à l'obtention d'une erreur donnée ne dépend pas de l'horizon de temps. Ces méthodes particulaires constituent une manière différente d'aborder la résolution numérique du problème de commande optimale stochastique qui semble, au vu des résultats présentés ici, mieux s'adapter aux problèmes multi-étapes.

Nous supposerons dans ce chapitre être dans un cadre markovien : les bruits affectant le système sont supposés indépendants d'un pas de temps à l'autre. Cela nous permet de nous placer dans le cadre classique de la commande optimale, et notamment de définir une notion d'état sur lequel la commande est en *feedback*. On peut voir l'état comme la statistique minimale permettant de prendre la décision optimale [2]. Ce n'est cependant pas une restriction théorique puisqu'il est toujours possible, lorsque l'on n'a pas indépendance temporelle des bruits, de définir l'état comme l'historique complet du processus de bruit. Nous avons déjà discuté ce cas dans le remarque 1.3 et reviendrons dessus en conclusion de ce chapitre.

Enfin, on trouve de nombreux travaux traitant de la manière de construire un arbre de

1. En particulier, un problème d'optimisation stochastique dont les contraintes et la fonction objectif sont linéaires reste linéaire lorsque l'on opère une discrétisation par arbres de scénarios. On peut alors utiliser les outils performants de résolution de programmes linéaires.

2. Il existe plusieurs définitions de la notion d'état. Nous nous référons ici à celle de Whittle (1982, Section 1.1).

scénarios. Dans le cas où on dispose d'un modèle théorique des aléas, on peut utiliser des tirages pseudo-aléatoires pour engendrer directement un arbre. Si, au contraire, on dispose seulement de chroniques des aléas, il s'agit de les rassembler pour former une structure arborescente ; cette question est abordée, entre autres, par Pflug (2001), Heitsch et Römisch (2003). La complexité du problème déterministe sur l'arbre dépend directement du nombre de nœuds présents dans celui-ci et il convient donc de penser au mieux la topologie de l'arbre afin d'effectuer la meilleure représentation possible du problème original sous cette contrainte de "budget" (nombre de nœuds limité). Nous mettons ici de côté ces questions pour nous concentrer sur le comportement asymptotique de la performance de l'arbre.

2.1 Évaluation de l'erreur

On rappelle le problème d'optimisation stochastique dynamique introduit au chapitre précédent. Soit $(\Omega, \mathcal{A}, \mathbb{P})$ un espace probabilisé. On considère trois types de variables aléatoires appartenant toutes à $L^2(\Omega, \mathcal{A}, \mathbb{P})$:

- $\boldsymbol{X} = (\boldsymbol{X}_t)_{t=0,\dots,T}$ que nous appelons *l'état*[3], et qui est à valeurs, à l'instant t, dans l'espace $\mathbb{X}_t = \mathbb{X} = \mathbb{R}^n$;
- $\boldsymbol{U} = (\boldsymbol{U}_t)_{t=0,\dots,T-1}$ que nous appelons *la commande*, et qui est à valeurs, à l'instant t, dans $\mathbb{U}_t = \mathbb{U} = \mathbb{R}^m$;
- $\boldsymbol{W} = (\boldsymbol{W}_t)_{t=0,\dots,T}$ que nous appelons *le bruit*, et qui est à valeurs, à l'instant t, dans $\mathbb{W}_t = \mathbb{W} = \mathbb{R}^p$.

Le bruit $\boldsymbol{W}$ est une donnée du problème, c'est-à-dire que sa loi de probabilité est connue, alors que $\boldsymbol{X}$ et $\boldsymbol{U}$ sont des variables d'optimisation.

On se place dans le cadre de la mémoire parfaite : à chaque instant t, l'aléa $\boldsymbol{W}_t$ est observé et "conservé", de sorte que l'information disponible est l'ensemble du passé du bruit $(\boldsymbol{W}_0, \dots, \boldsymbol{W}_t)$. Sur la base de cette information, une décision $\boldsymbol{U}_t$ est prise, ce qui induit un coût[4] $C_t(\boldsymbol{X}_t, \boldsymbol{U}_t)$. On introduit donc naturellement la tribu $\mathcal{A}_t = \sigma\{\boldsymbol{W}_0, \dots, \boldsymbol{W}_t\}$ par rapport à laquelle on demande à la commande d'être mesurable, ainsi que la filtration associée. L'état est ensuite transporté en $\boldsymbol{X}_{t+1} = f_t(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1})$. Plus précisément, on considère le problème de commande optimale stochastique suivant, qui est identique au problème (1.3) :

$$\min_{\boldsymbol{X},\boldsymbol{U}} \quad \mathbb{E}\left(\sum_{t=0}^{T} C_t(\boldsymbol{X}_t, \boldsymbol{U}_t)\right), \tag{2.1a}$$

$$\text{s.c.} \quad \boldsymbol{X}_{t+1} = f_t(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}), \qquad \forall t = 0, \dots, T-1, \tag{2.1b}$$

$$\boldsymbol{X}_0 = \boldsymbol{W}_0, \tag{2.1c}$$

$$\boldsymbol{U}_t \preceq \mathcal{A}_t. \tag{2.1d}$$

La contrainte (2.1d) est une contrainte de mesurabilité, appelée contrainte de non-anticipativité car elle indique que la commande $\boldsymbol{U}_t$ ne peut dépendre que des informations disponibles jusqu'à l'instant t et pas des réalisations des aléas futurs.

Supposons que le problème (2.1) ait une solution unique que nous noterons $(\boldsymbol{X}^*, \boldsymbol{U}^*)$. Supposons également que l'on dispose d'une méthode numérique, basée sur des tirages, qui fournit une solution approchée du problème (2.1) que nous noterons $(\boldsymbol{X}^\sharp, \boldsymbol{U}^\sharp)$.

3. Cette dénomination est abusive jusqu'à ce que nous faisions l'hypothèse 2.1.

4. Comme nous voyons plus loin, l'état est, par construction et du fait de la mesurabilité de $\boldsymbol{U}_t$, connu dès lors que l'on connaît le passé du bruit jusqu'à l'instant t.

Par la suite nous ferons l'hypothèse suivante (voir remarque 1.3 pour une justification de cette hypothèse) :

Hypothèse 2.1. Les variables aléatoires $\boldsymbol{W}_0, \dots, \boldsymbol{W}_T$ sont indépendantes.

Sous l'hypothèse 2.1, on sait, par le principe de programmation dynamique énoncé au §1.2.1, que la commande optimale est en *feedback* sur l'état $\boldsymbol{X}_t$. Autrement dit, il existe des fonctions γ_t^* telles que $\boldsymbol{U}_t^* = \gamma_t^*(\boldsymbol{X}_t^*)$, où $\boldsymbol{X}_0^* = \boldsymbol{W}_0$ et $\boldsymbol{X}_{t+1}^* = f_t(\boldsymbol{X}_t^*, \boldsymbol{U}_t^*, \boldsymbol{W}_{t+1})$, pour tout $t = 0, \dots, T-1$. Il est alors naturel de s'intéresser aux solutions approchées en *feedback* sur l'état $\boldsymbol{X}_t$. On construit donc, à partir de la solution approchée $(\boldsymbol{X}^\sharp, \boldsymbol{U}^\sharp)$, un *feedback* approché que l'on notera $\boldsymbol{\Gamma}_t^\sharp$, ce pour tout $t = 0, \dots, Te$. Nous précisons par la suite la manière de le construire.

Remarque 2.1. Notons que certaines méthodes numériques, telles que les arbres de scénarios, ne calculent pas directement des stratégies $\boldsymbol{\Gamma}_t^\sharp$ mais des réalisations du couple état-commande $(\boldsymbol{X}_t^\sharp, \boldsymbol{U}_t^\sharp)$. Se pose alors la question de la valeur de la commande en des valeurs de l'état non envisagées par la méthode. Autrement dit, comment obtenir une stratégie à partir de sa valeur en quelques points ? Pour ce faire, on aura naturellement recours à des opérateurs d'interpolation ou de régression. Cela sera précisé lors de la présentation de la méthode par arbres de scénarios.

Remarque 2.2. À la différence de γ^*, la stratégie approchée $\boldsymbol{\Gamma}^\sharp$ est une variable aléatoire. C'est le cas pour les algorithmes stochastiques, famille à laquelle appartiennent les méthodes d'arbres de scénarios et les méthodes particulaires que nous présentons au §2.2 et au §2.3. Pour ces algorithmes, la commande calculée dépend naturellement de l'état $\boldsymbol{X}_t$, mais dépend également des tirages effectués pour discrétiser la structure aléatoire. Ainsi, la stratégie solution obtenue lors d'une nouvelle expérience (pour laquelle on aura à faire un nouveau tirage de scénarios) sera différente de celles précédemment obtenues. Ce n'est pas le cas pour des algorithmes déterministes tels que la programmation dynamique, dont la solution ne varie pas d'une expérience à l'autre. Il convient de prendre en compte cette caractéristique dans l'évaluation de l'erreur.

Nous nous intéressons dans cette partie à l'évaluation de la "performance" de la solution approchée $\boldsymbol{\Gamma}^\sharp$. Il s'agit d'évaluer la distance entre cette solution approchée et la solution optimale, en fonction des différents paramètres associés à la méthode stochastique de résolution étudiée. Les stratégies sont ici des fonctions de l'état $\boldsymbol{X}_t$; afin d'évaluer des distances entre stratégies, il nous faut donc choisir une mesure sur l'espace d'état $\mathbb{X}$. La mesure qui paraît la plus naturelle est celle associée à la densité de l'état optimal, que nous noterons μ_t^* à l'instant t. On peut l'obtenir à partir de la stratégie solution γ^* en intégrant l'équation de Fokker-Planck (voir Quadrat, 2007, Section 1.2).

Nous pouvons maintenant introduire l'erreur quadratique moyenne (EQM) associée à la stratégie $\boldsymbol{\Gamma}^\sharp$, qui sera notre indicateur de performance :

$$\mathrm{EQM} = \mathbb{E}\left(\sum_{t=0}^{T} \int_{\mathbb{X}} \left\| \gamma_t^*(x) - \boldsymbol{\Gamma}_t^\sharp(x) \right\|^2 \mu_t^*(x)\, \mathrm{d}x \right). \tag{2.2}$$

Il est classique de décomposer l'EQM en deux termes que nous appellerons variance et carré du biais. Pour ce faire, nous avons besoin d'introduire une notation supplémentaire :

$\gamma_t^\sharp(\cdot) = \mathbb{E}\left(\mathbf{\Gamma}_t^\sharp(\cdot)\right)$ est l'espérance du *feedback* approché [5].

$$\mathrm{EQM} = \underbrace{\sum_{t=0}^{T} \int_{\mathbb{X}} \left\| \gamma_t^*(x) - \gamma_t^\sharp(x) \right\|^2 \mu_t^*(x)\,\mathrm{d}x}_{\text{Carré du biais}} + \underbrace{\mathbb{E}\left(\sum_{t=0}^{T} \int_{\mathbb{X}} \left\| \gamma_t^\sharp(x) - \mathbf{\Gamma}_t^\sharp(x) \right\|^2 \mu_t^*(x)\,\mathrm{d}x \right)}_{\text{Variance}}. \tag{2.3}$$

Ainsi, le carré du biais est la distance entre la stratégie optimale et la stratégie approchée moyenne, et la variance est la distance moyenne entre la stratégie approchée moyenne et une stratégie approchée particulière. Nous illustrons ces deux termes sur un exemple dans le cas de la méthode d'arbres de scénarios au §2.2.3 (Figure 2.2), et dans le cas des méthodes particulaires au §2.3.3 (Figure 2.4).

Remarque 2.3. Il faut remarquer que dans le cas d'un algorithme déterministe (la programmation dynamique par exemple), le terme de variance est nul par construction, comme nous l'avions déjà noté en d'autres termes dans la remarque 2.2. Il ne reste alors que le terme de biais dans l'EQM.

Dans la suite, nous évaluons l'EQM pour les arbres de scénarios ainsi que pour les méthodes particulaires et observons l'évolution de l'EQM en fonction des paramètres associés à ces méthodes. En particulier, nous nous intéressons à la relation asymptotique liant l'erreur au nombre de scénarios utilisés dans la méthode.

2.2 Arbres de scénarios

Pour les techniques arborescentes, nous avons déjà noté au §1.2.2 que les scénarios ne sont pas utilisés directement en optimisation mais sont liés sous la forme d'un arbre. Cela permet de représenter la contrainte de non-anticipativité sur les commandes : la décision ne peut dépendre que des observations passées, et pas des réalisations futures des variables aléatoires du problème. Ainsi, chaque nœud de l'arbre est lié à un passé unique (une seule branche mène de l'origine à ce nœud). En revanche, plusieurs sous-branches partent de celui-ci et représentent des "futurs possibles" des variables aléatoires du système. On fixe ici le nombre de sous-branches partant de chaque nœud de l'arbre à une quantité constante, appelée taux de bifurcation. Les scénarios "fils" sont utiles afin de calculer de manière approchée, par Monte-Carlo, les grandeurs statistiques qui interviennent dans le problème de commande optimale, notamment, dans notre cas, les espérances conditionnelles. Or, l'évaluation par Monte-Carlo de l'espérance conditionnelle d'une variable aléatoire définie sur le pas de temps directement consécutif au nœud courant dépendra du nombre de tirages utilisés, soit en l'occurrence du nombre de fils de ce nœud, autrement dit du taux de bifurcation. On s'attend donc à ce que l'erreur soit liée non pas au nombre total de scénarios échantillonnés afin de construire l'arbre, mais au taux de bifurcation. Ainsi, pour garantir une précision donnée, il faudra fixer un certain taux de bifurcation et le nombre de scénarios nécessaires à cette opération variera donc de manière exponentielle avec l'horizon de temps du problème.

5. Cette espérance porte sur les tirages de variables aléatoires effectués au cours de l'algorithme d'où provient la stratégie approchée $\mathbf{\Gamma}^\sharp$.

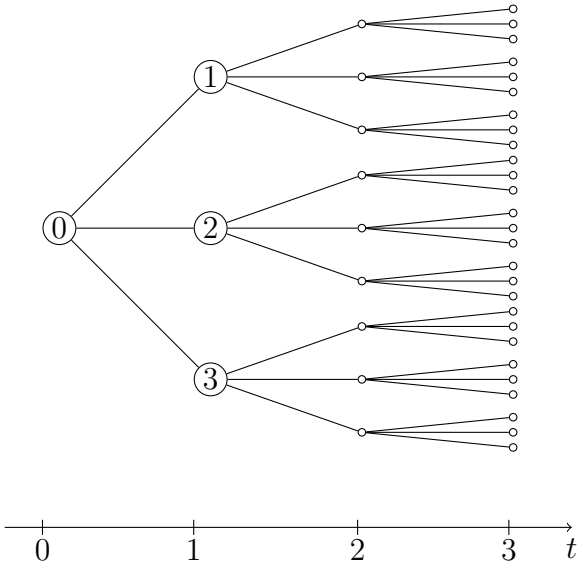

FIGURE 2.1 – Exemple de structure arborescente

2.2.1 Présentation succincte

Afin de résoudre numériquement les problèmes de commande optimale stochastique, parmi lesquels se trouve le problème (2.1), un certain nombre de méthodes, souvent rassemblées sous le terme *Stochastic Programming* (voir Ruszczyński et Shapiro, 2003), proposent de discrétiser l'aléa sous la forme d'un arbre de scénarios afin de se ramener à un problème d'optimisation dans lequel les variables aléatoires ont été échantillonnées et sur lequel on saura appliquer des techniques d'optimisation classique, éventuellement de décomposition. Il faut noter que cela constitue plus une méthodologie qu'un algorithme au sens où ces méthodes, que nous qualifierons ici de méthodes d'arbres de scénarios, proposent une façon de se ramener à un problème plus simple pour lequel on ne précise pas, en général, l'algorithme qui permettra de le résoudre. Contrairement aux méthodes particulaires que nous décrirons ensuite, nous sommes ici face à une approche "discrétisation-optimisation" : on discrétise d'abord le problème, puis on écrit les conditions d'optimalité associées, et on les résout.

L'idée de base est la suivante : représenter la "diffusion de l'information" sous la forme d'une structure arborescente, dont nous donnons une illustration dans la figure 2.1. Plaçons-nous en $t = 0$ et supposons que nous connaissons la réalisation w_0 de $\boldsymbol{W}_0$; on a ainsi une seule racine pour notre arbre (nœud 0). On considère alors trois possibilités pour les réalisations futures de l'aléa $\boldsymbol{W}_1$, ce que nous représentons par trois successeurs au nœud d'origine dans l'arbre (nœuds 1, 2, et 3). Pour chacun d'entre eux, on effectue un tirage de la variable aléatoire $\boldsymbol{W}_1$. La procédure de construction se poursuit ainsi jusqu'à arriver à la fin de l'horizon du problème. Notons que l'on a choisi de prendre une seule racine, mais que nous aurions pu en considérer plusieurs ; ce sera le cas dans l'exemple numérique du §2.2.3.

Notons $\mathcal{N}$ l'ensemble des nœuds, $\mathcal{R}$ l'ensemble des racines et $\mathcal{L}$ l'ensemble des feuilles de l'arbre. Une fois cet arbre construit, on a en fait défini les fonctions suivantes :

- la fonction temps $\theta \ : \mathcal{N} \longrightarrow \{0, 1, \ldots, T\}$ qui à tout nœud associe la pas de temps associé et par là même la multi-application θ^{-1} qui à tout pas de temps $t \in \{0, \ldots, T\}$ associe l'ensemble des nœuds du niveau t ;

- la fonction père $\nu : \mathcal{N}\backslash\mathcal{R} \longrightarrow \mathcal{N}\backslash\mathcal{L}$ qui à un nœud associe le nœud qui le précède dans l'arbre ;
- la fonction poids $\pi : \mathcal{N} \longrightarrow [0,1]$ qui à tout nœud de l'arbre associe la probabilité de passer par ce nœud (on a donc que la somme des $\pi(i)$ pour $i \in \theta^{-1}(t)$ est égale à 1, pour tout t) ;
- la multi-application $F = \nu^{-1}$ qui à chaque nœud i de l'arbre associe l'ensemble de ses fils (on adopte la convention que si i est terminal dans l'arbre, alors $F(i) = \emptyset$) ;
- la fonction F^* qui à tout nœud i associe le sous-arbre pendant à i, soit $F^*(i) = F(i) \cup F^2(i) \cup \cdots \cup F^T(i)$.

À l'aide de ces fonctions, il nous est maintenant possible d'écrire le pendant discrétisé du problème (2.1) :

$$\min_{x,u} \quad \sum_{i\in\mathcal{N}} \pi(i) \cdot C_{\theta(i)}\left(x_i, u_i\right), \tag{2.4a}$$

$$\text{s.c.} \quad x_i = f_{\theta(\nu(i))}\left(x_{\nu(i)}, u_{\nu(i)}, w_i\right), \qquad \forall i \in \mathcal{N}\backslash\mathcal{R}, \tag{2.4b}$$

$$x_i = w_i, \qquad \forall i \in \mathcal{R}. \tag{2.4c}$$

Dans la pratique, on dispose souvent d'un ensemble de scénarios supposés indépendants et il convient de construire une structure arborescente à partir de ces scénarios de manière optimale, en un sens à définir. Il existe une littérature abondante concernant la manière de construire les arbres de scénarios (Pflug, 2001, Heitsch et Römisch, 2003), ainsi que sur la stabilité du problème d'optimisation résultant vis-à-vis de cette discrétisation (Heitsch *et al.*, 2006). Nous ne nous intéresserons pas ici à ce sujet et nous supposerons que l'arbre a été construit avec un taux de branchement constant n_b, à partir de tirages indépendants, ce qui conduit à n_b^T feuilles, ou encore $1 + n_b + \cdots + n_b^T = (n_b^{T+1} - 1)/(n_b - 1)$ nœuds dans l'arbre. Ainsi, dans l'exemple de la figure 2.1, on a choisi un taux de branchement $n_b = 3$, ce qui nous conduit à $3^3 = 27$ feuilles dans l'arbre. On se rend compte qu'afin d'appliquer une telle méthode, il est avant toute chose nécessaire d'avoir à notre disposition un nombre de scénarios suffisamment important pour pouvoir construire une telle structure avec un taux de branchement raisonnable (ici il nous faut au moins 27 scénarios).

Insistons à nouveau sur le fait que le problème ainsi discrétisé n'est plus stochastique : on a tiré des échantillons des variables aléatoires $\boldsymbol{W}_0, \ldots, \boldsymbol{W}_T$. Comme nous l'avions précédemment noté (Remarque 2.2), la solution du problème discrétisé dépend par construction des tirages effectués lors de la construction de l'arbre. Il convient de garder cela à l'esprit pour l'étude de l'erreur.

2.2.2 Erreur dans un cadre général

Nous pouvons maintenant nous intéresser à l'évaluation de notre indicateur de performance, l'EQM, pour les méthodes d'arbres de scénarios. Il existe dans la littérature peu de résultats traitant de la vitesse de convergence de telles méthodes pour les problèmes à plusieurs pas de temps. Le plus précis est sans doute celui de Shapiro, que nous rappelons par la suite. En utilisant des outils issus de la théorie des grandes déviations, il montre que le nombre de scénarios (soit le nombre de feuilles dans l'arbre) nécessaire à l'obtention d'une précision donnée augmente de façon exponentielle avec l'horizon de temps, comme cela était couramment constaté en pratique. En effet, la principale source d'erreur [6] provient de l'évaluation des espérances conditionnelles du coût d'un pas de temps sur l'autre,

6. Celle-ci n'est pas définie de la même manière que dans notre cas. Nous préciserons sa nature par la suite.

qui est effectuée à l'aide de tirages du bruit ; elle évolue donc, en tant qu'estimateur de Monte-Carlo, comme l'inverse du nombre de tirages utilisés, qui correspond ici au taux de branchement et non au nombre de scénarios.

Résultat asymptotique en terme de grandes déviations

Nous rappelons ici le résultat de Shapiro (2006), obtenu pour un problème à trois pas de temps. L'étude se généralise directement à un problème à T pas de temps, comme il est précisé dans l'article. L'auteur choisit d'étudier un problème à seulement trois pas de temps car cela est suffisant pour mettre en évidence les caractéristiques des problèmes de commande optimale stochastique. La même étude étendue à plus de trois pas de temps serait fastidieuse du point de vue des notations, sans apporter d'éléments supplémentaires à la discussion.

La formulation et les notations de l'article sont quelque peu différentes des nôtres et correspondent aux notations les plus largement admises dans la littérature *Stochastic Programming*. Ainsi, partant d'un état initial dans $\mathcal{X}_1 \subset \mathbb{R}^{n_1}$, l'auteur n'introduit pas explicitement de contrôle mais considère qu'à chaque pas de temps $t < T$ et pour chaque état donné x_t, il existe un espace $\mathcal{X}_{t+1}(x_t, \boldsymbol{\xi}_{t+1}) \subset \mathbb{R}^{n_{t+1}}$ dans lequel on peut choisir la valeur de l'état x_{t+1}. La commande $\boldsymbol{U}$ que nous introduisons habituellement n'est présente ici que de manière implicite. Vu de l'instant t, on peut identifier la variable x_t à l'état "habituel" du système ; la variable x_{t+1}, en revanche, représente à la fois l'état à l'instant suivant et la décision prise à l'instant t. Nous confondrons donc parfois, dans cette partie, décision et état. Le problème de commande est formulé ainsi :

$$\min_{x_1 \in \mathcal{X}_1} F_1(x_1) + \mathbb{E}\left(\min_{x_2 \in \mathcal{X}_2(x_1, \boldsymbol{\xi}_2)} F_2(x_2, \boldsymbol{\xi}_2) + \mathbb{E}\left(\cdots + \mathbb{E}\left(\min_{x_T \in \mathcal{X}_T(x_{T-1}, \boldsymbol{\xi}_T)} F_T(x_T, \boldsymbol{\xi}_T)\right)\right)\right). \tag{2.5}$$

La première décision x_1 est en boucle ouverte, c'est-à-dire qu'elle est prise avant toute observation du système. Ce n'est donc pas une variable aléatoire. Suite à cette prise de décision, un aléa $\boldsymbol{\xi}_2$ se réalise, est observé et la décision x_2 est prise au regard de cette observation. L'ensemble admissible $\mathcal{X}_2$ est paramétré à la fois par la première décision x_1 et par $\boldsymbol{\xi}_2$. Il faut comprendre que même si, conditionnellement à l'observation de $\boldsymbol{\xi}_2$, la deuxième décision x_2 n'est pas une variable aléatoire (nous ne l'écrivons volontairement pas en lettres grasses), il n'en reste pas moins qu'elle est indexée par la réalisation de $\boldsymbol{\xi}_2$: on est donc bien dans un modèle de décision du type "boucle fermée". La première espérance dans (2.5) porte sur la variable aléatoire $\boldsymbol{\xi}_2$, la suivante sur $\boldsymbol{\xi}_3|\boldsymbol{\xi}_2$, et ainsi de suite jusqu'à la dernière espérance, qui porte sur la variable aléatoire $\boldsymbol{\xi}_T|\boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_{T-1}$. Dans cette formulation, la contrainte de non-anticipativité est donc écrite "en dur" dans le problème : la décision x_t est prise conditionnellement à la connaissance de $\boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_t$.

Nous avons également besoin d'introduire des fonctions valeur indexées par le temps et représentant, à la manière d'une fonction de Bellman, le coût optimal futur connaissant, à l'instant t, la valeur x_{t-1} de l'état précédent et celle ξ_t du bruit :

$$Q_t(x_{t-1}, \xi_t) = \min_{x_t \in \mathcal{X}_t(x_{t-1}, \xi_t)} F_t(x_t, \xi_t) + \mathbb{E}\left(\cdots + \mathbb{E}\left(\min_{x_T \in \mathcal{X}_T(x_{T-1}, \boldsymbol{\xi}_T)} F_T(x_T, \boldsymbol{\xi}_T)\right)\right).$$

Avant de présenter le résultat de convergence, il est important de montrer que la formulation que nous venons de présenter correspond bien à celle introduite dans le chapitre 1. Afin que le problème (1.3) rentre dans le formalisme de (2.5), il suffit de poser comme

nouvel état

$$\boldsymbol{Y}_t := (\boldsymbol{X}_t, \boldsymbol{U}_t),$$

ainsi que

$$\mathcal{X}_{t+1}\left(\boldsymbol{Y}_t, \boldsymbol{\xi}_{t+1}\right) := \Big\{\boldsymbol{Y}_{t+1} = (\boldsymbol{X}_{t+1}, \boldsymbol{U}_{t+1}), \\ \text{t.q. } \boldsymbol{X}_{t+1} = f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{\xi}_{t+1}\right), \boldsymbol{Y}_t = (\boldsymbol{X}_t, \boldsymbol{U}_t)\Big\},$$

et

$$F_t\left(\boldsymbol{Y}_t, \boldsymbol{\xi}_t\right) := C_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t\right).$$

La discrétisation opérée sur le problème (2.5) est en tout point la même que celle décrite au §1.2.2 et rappelée au §2.2.1. On utilise un taux de branchement $n_{b,t}$ au pas de temps t. Un seul nœud est à l'origine de l'arbre puisque la décision à $t = 1$ est en boucle ouverte. Au pas de temps $t = 2$, on effectue $n_{b,2}$ tirages $\xi_2^1, \ldots, \xi_2^{n_{b,2}}$ de la variable aléatoire $\boldsymbol{\xi}_2$. Pour chaque réalisation ξ_2^i, on réalise, au pas de temps $t = 3$, $n_{b,3}$ tirages de la variable aléatoire $\boldsymbol{\xi}_3|\boldsymbol{\xi}_2 = \xi_2^i$. Au total, nous avons donc besoin de $n_{b,2} \times n_{b,3}$ scénarios pour construire l'arbre.

Nous nous intéressons à la qualité de la première décision issue de l'arbre vis-à-vis du problème original. Afin d'énoncer le résultat, nous avons besoin de la définition suivante.

Définition 2.2 (ε-optimalité)**.** Soit J une fonction d'un espace de Hilbert $\mathbb{X}$ dans $\mathbb{R}$. On dit que x^ε est une solution ε-optimale du problème consistant à minimiser J que $\mathbb{X}$ si :

$$J\left(x^\varepsilon\right) - \inf_{x \in \mathbb{X}} J\left(x\right) < \varepsilon.$$

Considérons le problème le plus extérieur dans l'expression (2.5), c'est-à-dire le problème de minimisation en x_1, ainsi que le problème le plus extérieur du pendant discrétisé par Monte-Carlo du problème (2.5). Le résultat de Shapiro consiste à donner des conditions permettant de relier l'ensemble des solutions ε-optimales du problème discrétisé et l'ensemble des solutions ε-optimales du problème continu.

Nous aurons besoin des hypothèses suivantes :

Hypothèse 2.3. Les aléas $\boldsymbol{\xi}_2$ et $\boldsymbol{\xi}_3$ sont indépendants.

Cette hypothèse est utile pour simplifier la mise en œuvre de la preuve. De plus, les espérances conditionnelles intervenant dans le problème se réduisent alors à des espérances, ce qui simplifie également la mise en œuvre de l'approche elle-même. Les hypothèses qui suivent sont d'ordre technique.

Hypothèse 2.4. L'ensemble $\mathcal{X}_1$ a un diamètre fini D_1.

Hypothèse 2.5. Il existe une constante $L_1 > 0$ telle que

$$\left|Q_2\left(x_1', \boldsymbol{\xi}_2\right) - Q_2\left(x_1, \boldsymbol{\xi}_2\right)\right| \leq L_1 \left\|x_1' - x_1\right\|$$

pour tous $x_1, x_1' \in \mathcal{X}_1$ et presque tout $\boldsymbol{\xi}_2$.

Hypothèse 2.6. Il existe une constante $\sigma_1 > 0$ telle que pour tout $x_1 \in \mathcal{X}_1$:

$$M_{1,x_1}(t) \leq \exp\left(\frac{\sigma_1^2 t^2}{2}\right), \qquad \forall t \in \mathbb{R},$$

où $M_{1,x_1}(t)$ est la fonction génératrice des moments de $Q_2(x_1, \boldsymbol{\xi}_2) - \mathbb{E}\left(Q_2(x_1, \boldsymbol{\xi}_2)\right)$.

Hypothèse 2.7. Il existe une constante positive D_2 telle que pour tout $x_1 \in \mathcal{X}_1$ et presque tout $\boldsymbol{\xi}_2$ l'ensemble $\mathcal{X}_2(x_1, \boldsymbol{\xi}_2)$ ait un diamètre fini inférieur ou égal à D_2.

Hypothèse 2.8. Il existe une constante $L_2 > 0$ telle que

$$\left|F_2\left(x_2', \boldsymbol{\xi}_2\right) - F_2\left(x_2, \boldsymbol{\xi}_2\right) + Q_3\left(x_2', \boldsymbol{\xi}_3\right) - Q_3\left(x_2, \boldsymbol{\xi}_3\right)\right| \leq L_2 \left\|x_2' - x_2\right\|$$

pour tous $x_2', x_2 \in \mathcal{X}_2(x_1, \boldsymbol{\xi}_2), x_1 \in \mathcal{X}_1$ et presque tout $\boldsymbol{\xi}_2$ et $\boldsymbol{\xi}_3$.

Hypothèse 2.9. Il existe une constante $\sigma_2 > 0$ telle que pour tout $x_2 \in \mathcal{X}_2(x_1, \boldsymbol{\xi}_2)$, tout $x_1 \in \mathcal{X}_1$ et presque tout $\boldsymbol{\xi}_2$:

$$M_{2,x_2}(t) \leq \exp\left(\frac{\sigma_2^2 t^2}{2}\right), \qquad \forall t \in \mathbb{R},$$

où $M_{2,x_2}(t)$ est la fonction génératrice des moments de $Q_3(x_2, \boldsymbol{\xi}_3) - \mathbb{E}\left(Q_3(x_2, \boldsymbol{\xi}_3)\right)$.

Nous pouvons à présent citer le résultat de Shapiro.

Théorème 2.10 (Shapiro, 2006, Théorème 2)**.** *On se place sous les hypothèses 2.3 à 2.9. Soit, de plus, $\varepsilon > 0$, $\alpha \in]0,1[$. Si, enfin, $n_{b,1}$ et $n_{b,2}$ sont tels que*

$$\mathcal{O}(1)\left[\left(\frac{D_1 L_1}{\varepsilon}\right)^{n_1} \exp\left(-\frac{\mathcal{O}(1) n_{b,1} \varepsilon^2}{\sigma_1^2}\right) + \left(\frac{D_1 L_3}{\varepsilon}\right)^{n_1}\left(\frac{D_2 L_2}{\varepsilon}\right)^{n_2} \exp\left(-\frac{\mathcal{O}(1) n_{b,2} \varepsilon^2}{\sigma_2^2}\right)\right] \leq \alpha.$$

Alors on a que toute solution $\frac{\varepsilon}{2}$-optimale pour le problème discrétisé par arbres de scénarios est une solution ε-optimale pour (2.5) *avec probabilité au moins égale à $1 - \alpha$.*

Si on considère un taux de branchement constant $n_b = n_{b,1} = n_{b,2}$ et si on pose $L := \max(L_1, L_2, L_3)$, $D := \max(D_1, D_2)$ et $\sigma := \max(\sigma_1, \sigma_2)$ on obtient l'estimation suivante pour le taux de branchement nécessaire à l'obtention d'une précision ε avec probabilité $1 - \alpha$:

$$\mathcal{O}(1)\left(\frac{DL}{\varepsilon}\right)^{n_1+n_2} \exp\left(-\frac{\mathcal{O}(1) n_b \varepsilon^2}{\sigma^2}\right) \leq \alpha,$$

ce qui revient à

$$n_b \geq \frac{\mathcal{O}(1)\sigma^2}{\varepsilon^2}\left[(n_1 + n_2)\log\left(\frac{DL}{\varepsilon}\right) + \log\left(\frac{\mathcal{O}(1)}{\alpha}\right)\right]. \tag{2.6}$$

Naturellement, le taux de branchement doit croître lorsque la précision demandée augmente (c'est-à-dire lorsque ε diminue) et lorsque la probabilité avec laquelle on demande que la solution discrétisée réalise effectivement cette précision augmente (c'est-à-dire lorsque α diminue).

Mais le résultat important est le suivant : la précision associée au schéma de discrétisation par arbres de scénarios est ici reliée au taux de branchement. Ainsi, il semble que pour un problème à quatre pas de temps, nous aurons besoin de n_b fois plus de scénarios pour construire l'arbre nécessaire à l'obtention de la même précision que pour le problème à trois pas de temps. Si on itère ce raisonnement, on obtient donc que le nombre de scénarios nécessaire à l'obtention d'une certaine précision croît de manière exponentielle avec l'horizon de temps, à la manière de n_b^T !

2.2.3 Exemple d'un problème de filtrage

Nous étudions ici, pour un exemple simple de commande optimale stochastique où l'espace d'état est de dimension 1, la vitesse à laquelle l'EQM de la solution approchée par méthode d'arbres de scénarios converge vers la solution optimale. En ce qui concerne la structure de l'arbre, nous considérons une répartition régulière des branches, c'est-à-dire qu'un unique paramètre, noté n_b, indique le nombre de fils de chaque nœud de l'arbre. C'est en fonction de ce paramètre que nous nous intéressons à la convergence de l'EQM.

On considère le problème suivant :

$$\min_{\boldsymbol{U}_0,\dots,\boldsymbol{U}_{T-1}} \quad \mathbb{E}\left(\varepsilon \sum_{t=0}^{T-1} \boldsymbol{U}_t^2 + \boldsymbol{X}_T^2\right), \tag{2.7a}$$

$$\text{s.c.} \quad \boldsymbol{X}_{t+1} = \boldsymbol{X}_t + \boldsymbol{U}_t + \boldsymbol{W}_{t+1}, \quad \forall t = 0,\dots,T-1, \tag{2.7b}$$

$$\boldsymbol{X}_0 = \boldsymbol{W}_0, \tag{2.7c}$$

$$\boldsymbol{U}_t \preceq \boldsymbol{W}_0,\dots,\boldsymbol{W}_t, \quad \forall t = 0,\dots,T-1. \tag{2.7d}$$

Les variables aléatoires $\boldsymbol{W}_0,\dots,\boldsymbol{W}_T$ sont i.i.d. de loi uniforme sur $[-1,1]$. On montre facilement que la solution de ce problème s'écrit $\boldsymbol{U}_t^* = \gamma_t^*(\boldsymbol{X}_t^*)$ avec :

$$\gamma_t^*(x) = -\frac{x}{T-t+\varepsilon}. \tag{2.8}$$

Nous allons maintenant résoudre le problème discrétisé à l'aide d'un arbre de scénarios. En chaque nœud i de l'arbre, on a un tirage du bruit $\boldsymbol{W}_i^\sharp$ et la résolution du problème sur l'arbre nous donne une valeur pour la commande $\boldsymbol{U}_i^\sharp$ et une valeur pour l'état $\boldsymbol{X}_i^\sharp$. Ainsi, on montre facilement que :

$$\boldsymbol{U}_i^\sharp = -\frac{\boldsymbol{X}_i^\sharp + \left(\sum_{j\in F^*(i)} \boldsymbol{W}_j^\sharp\right) \Big/ n_b^{T-\theta(i)}}{T-\theta(i)+\varepsilon}, \tag{2.9}$$

où la somme au numérateur correspond à une espérance discrétisée sur le sous-arbre partant du nœud i. Afin d'obtenir une commande pour toute valeur de l'état, on construit à chaque pas de temps des cellules de Voronoï $\boldsymbol{C}_i^\sharp$ centrées en les valeurs $\boldsymbol{X}_i^\sharp$ de l'état, sur lesquelles la stratégie prendra la valeur $\boldsymbol{U}_i^\sharp$. On obtient ainsi une stratégie de la forme :

$$\boldsymbol{\Gamma}_t^\sharp(x) = \sum_{i\in\theta^{-1}(t)} \boldsymbol{U}_i^\sharp \mathbf{1}_{\boldsymbol{C}_i^\sharp}(x).$$

La fonction $\boldsymbol{\Gamma}_t^\sharp$ est une variable aléatoire, car elle dépend des tirages effectués pour construire l'arbre. Elle dépend également de l'opérateur d'interpolation choisi pour évaluer cette stratégie en des valeurs de l'état ne correspondant pas à l'une des valeurs $\boldsymbol{X}_i^\sharp$ obtenues à l'issue de la résolution du problème sur l'arbre. Nous avons choisi ici d'utiliser un opérateur de plus proche voisin. Bien que celui-ci compte parmi les opérateurs d'interpolation les plus frustres, il semble, au regard des résultats que nous présentons plus loin, suffisant afin de tirer nos conclusions.

On présente dans les graphiques de la figure 2.2 les différentes commandes qui interviennent dans le calcul de l'EQM. Ainsi, on représente dans l'encart de gauche la stratégie optimale γ_0^* et la stratégie approchée moyenne $\gamma_0^\sharp$ [7]. Dans l'encart de droite, on trace plusieurs tirages de la stratégie $\boldsymbol{\Gamma}_0^\sharp$. On a choisi ici un taux de branchement $n_b = 3$. De plus,

7. telle que $\gamma_0^\sharp(x) = \mathbb{E}\left(\boldsymbol{\Gamma}_0^\sharp(x)\right)$, $\forall x \in \mathbb{X}$

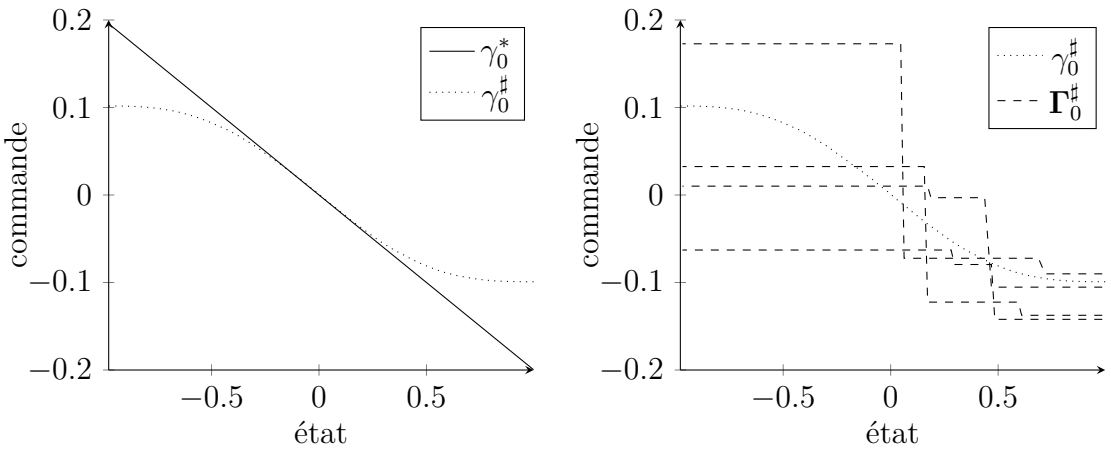

FIGURE 2.2 – *Feedbacks* exacts et approchés issus d'une méthode par arbre de scénarios.

on a choisi $\varepsilon = 1$, $T = 4$, et le nombre d'expériences effectuées afin d'approcher les espérances présentes dans le calcul de l'EQM (en particulier, la commande moyenne présentée dans la figure 2.2) est fixé à 10^4. Il en est de même pour l'estimation de la vitesse de convergence que nous présentons ensuite.

Dans l'encart de droite, on observe bien la structure par paliers des stratégies approchées par arbres de scénarios, due à notre choix d'un interpolateur constant par morceaux pour la reconstruction de stratégies à partir des valeurs de la commande en les nœuds de l'arbre. Cependant, on observe que moyenner les stratégies approchées a un effet de lissage sur la commande. Ainsi, même si chaque stratégie est constante par morceaux, la stratégie moyenne semble continue, voire lisse. C'est assez naturel : les frontières des paliers sont également des variables aléatoires dépendant de l'arbre tiré. Si la fonction à approcher est suffisamment lisse, soient deux points proches dans l'espace d'état, ceux-ci se retrouveront, si on répète un grand nombre de fois l'expérience, un grand nombre de fois dans la même cellule, et donc avec la même valeur de commande.

Naturellement, l'erreur commise par la méthode d'arbres est plus importante sur les bords de l'intervalle car la probabilité qu'un point situé sur le bord gauche, par exemple, ait le plus proche voisin à sa droite est supérieure à la probabilité que le plus proche voisin soit sur sa gauche. Moyenné sur un grand nombre d'expériences, cela se traduit par un biais plus important sur les bords de l'intervalle.

Nous passons maintenant à l'évaluation de la vitesse de convergence de l'erreur, c'est-à-dire à la façon dont l'EQM varie lorsque l'on fait varier le taux de branchement n_b. Nous évaluons l'EQM, soit la distance entre la stratégie optimale γ^* et la stratégie approchée $\boldsymbol{\Gamma}^\sharp$. Le calcul analytique de ces quantités apparaît difficile, notamment du fait des interpolations nécessaires à la synthèse de *feedback* à partir des décisions obtenues en les nœuds de l'arbre. Cependant, il est assez facile de les évaluer numériquement, en répétant un grand nombre de fois l'expérience consistant à tirer un arbre d'aléas et à résoudre le problème ainsi discrétisé afin d'évaluer les espérances apparaissant dans (2.3). Ainsi adopte-t-on le protocole expérimental suivant :

1. on opère n_b tirages du bruit $\boldsymbol{W}_0$, n_b^2 tirages de $\boldsymbol{W}_1$, …, n_b^{T+1} tirages de $\boldsymbol{W}_T$;
2. on calcule en chaque nœud i de l'arbre les valeurs de l'état et de la commande qui en découlent à l'aide de la relation (2.9) ;
3. pour chaque pas de temps, on "étend" ces commandes à tout l'espace d'état (et pas

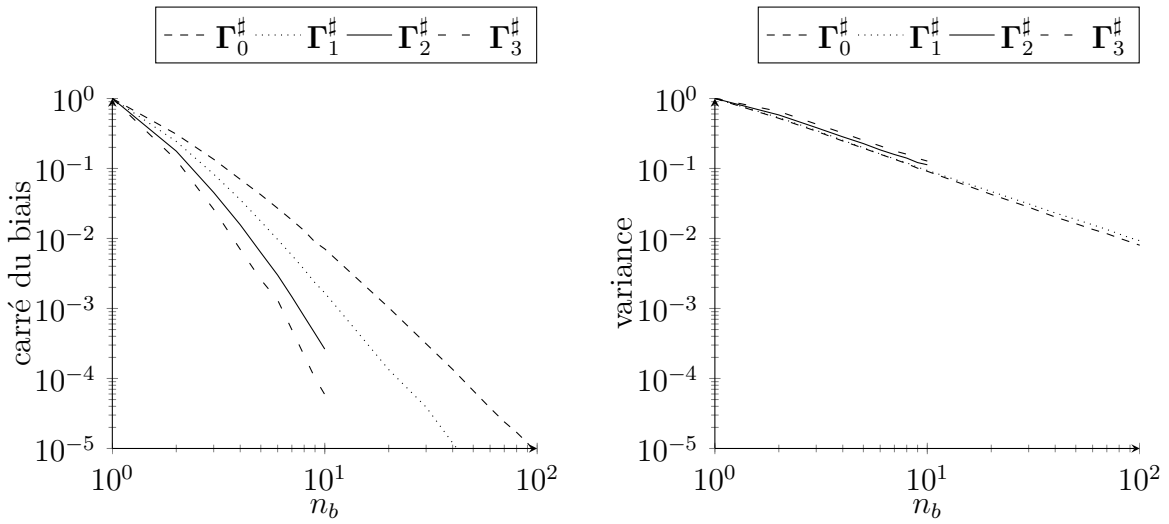

FIGURE 2.3 – Carré du biais et variance de la commande par arbres de scénarios en fonction du taux de branchement.

seulement au nœud où elles ont été calculées) en construisant le pavage de Voronoï associé aux valeurs de l'état obtenues ;

4. on stocke la stratégie fonction de l'état ainsi obtenue.

On répète cette expérience, de manière indépendante d'une expérience à l'autre, afin de pouvoir approcher les espérances présentes dans la relation (2.3). Nous estimons ainsi le biais et la variance des stratégies "arbres de scénarios" pour un taux de branchement n_b variant de 1 à 10 au moins. Le nombre de calcul à effectuer pour la stratégie au pas de temps t évoluant comme n_b^t, l'expérience prend trop de temps pour $n_b > 10$ dans le cas de $t = 4$. Pour les pas de temps inférieurs, on peut mener l'expérience un peu plus loin.

On obtient les courbes de biais et de variance de la figure 2.3. On a représenté sur le graphique de gauche l'évolution du carré du biais comme fonction du taux de branchement n_b de l'arbre, pour chacune des T stratégies. On fait de même pour ce qui concerne la variance sur le graphique de droite.

On observe clairement que le terme dominant de l'EQM est la variance, qui décroît comme l'inverse du taux de branchement n_b, et ce quel que soit le pas de temps. Le carré du biais, quant à lui, décroît au moins aussi rapidement que $n_b^{-2.5}$, et ce taux de décroissance varie en fonction du pas de temps, devenant de plus en plus important avec le temps. En effet, la structure d'arbre fait que le nombre de nœuds à un pas de temps donné augmente avec le temps. On a ainsi, avec le temps, de plus en plus de points pour représenter la stratégie, dont la dimension de l'espace de départ (l'espace d'état) ne varie pas avec le temps. Il est donc naturel d'observer un biais de meilleure qualité pour les stratégies approchées aux derniers pas de temps.

Nous venons d'observer qu'afin de multiplier la précision de la solution par un facteur α il fallait multiplier le taux de branchement n_b par ce même facteur α, et donc le nombre de nœuds dans l'arbre par un facteur α^{T+1} ! Autrement dit, la taille du problème discrétisé nécessaire à l'obtention d'une précision donnée évolue de manière exponentielle en fonction de l'horizon de temps. Dans l'exemple considéré, nous ne sommes ainsi pas capables de calculer biais et variance pour les stratégies des derniers pas de temps dès lors que le taux de branchement dépasse 10, car le nombre de nœuds impliqués devient trop important.

Notons que ce résultat numérique corrobore le constat de Shapiro (2006), bien que son étude portait sur la distance entre les coûts optimaux au sens des grandes déviations, et non sur la distance entre stratégies : il semble qu'il soit en général numériquement difficile, pour des problèmes à plusieurs pas de temps, d'obtenir une précision satisfaisante à l'issue de la discrétisation par arbres de scénarios, du fait que le nombre de nœuds requis croît exponentiellement avec l'horizon de temps, atteignant rapidement une taille que nous ne saurions gérer numériquement.

Nous n'avons ici considéré que le cas d'un problème où la dimension de l'état est 1. Au §2.4, nous étudierons un cas en dimension supérieure, afin d'observer la manière dont l'erreur associée à la discrétisation par arbres de scénarios évolue en fonction de ce paramètre.

2.3 Méthode particulaire

Nous venons de constater sur un exemple que la discrétisation du problème (2.1) sous la forme d'un arbre de scénarios entraînait une erreur sur la solution qui décroît comme l'inverse du taux de branchement de l'arbre, indépendamment de l'horizon de temps. Cela implique que le nombre de scénarios nécessaire à l'obtention d'une précision donnée augmente de manière exponentielle avec l'horizon de temps du problème et rend cette méthodologie difficilement utilisable pour les problèmes qui nous intéressent, où le nombre de pas de temps est souvent au moins de l'ordre de plusieurs dizaines.

Nous montrons dans cette partie que, sous certaines hypothèses, les méthodes particulaires introduites dans la thèse de Dallagi (2007) permettent d'éviter cette explosion de la complexité en fonction de l'horizon de temps. On marie ici des idées se rapprochant, d'une part, de la méthode de Monte-Carlo pour l'estimation d'espérance (discrétisation de l'aléa sous forme de scénarios), d'autre part, de la programmation dynamique (introduction d'un état et utilisation de la propriété de Markov). Dans cette partie, nous nous intéressons à la version appelée "espérance du gradient, espérance de l'état adjoint", que nous avons déjà évoqué au chapitre 1. Dans celle-ci, on a éliminé toute espérance conditionnelle et il ne nous reste plus que des calculs d'espérance à effectuer. Comme pour les méthodes arborescentes, les méthodes particulaires utilisent des tirages des aléas, ou scénarios, afin de discrétiser la structure probabiliste du problème.

Le cheminement est le même qu'au §2.2 : après un rappel de la présentation des méthodes particulaires faites au chapitre 1, nous cherchons à évaluer l'erreur commise par cette nouvelle discrétisation en fonction du nombre de scénarios utilisés. Nous montrons les difficultés de l'étude théorique de cette erreur dans un cadre général et proposons une évaluation numérique, sur le même problème de filtrage qu'au §2.2.3, de l'évolution de l'EQM en fonction du nombre de scénarios. Nous montrons que l'erreur est maintenant liée au nombre de scénarios utilisés et que le nombre de scénarios nécessaires à l'obtention d'une précision donnée ne dépend pas de l'horizon de temps, contrairement à ce que nous avions observé avec les arbres de scénarios.

Il faut cependant garder à l'esprit que les expériences numériques que nous menons jusque dans cette partie concernent un problème où l'espace d'état est de dimension 1. Or, une des difficultés majeures auxquelles sont confrontées les méthodes numériques associées aux problèmes de commande optimale stochastique est l'augmentation de la dimension de l'espace d'état (*malédiction de la dimension*). Il conviendra donc d'évaluer, pour chacune des deux méthodes étudiées ici, leur comportement sur des problèmes de plus grande dimension. Ce travail sera effectué au §2.4.

2.3.1 Présentation succincte

Les méthodes particulaires[8] consistent, contrairement aux arbres de scénarios, à écrire des conditions d'optimalité avant de discrétiser l'aléa : il s'agit d'une méthode du type "optimisation-dicrétisation". Elles sont naturellement de nature variationnelle au sens où, à travers les conditions d'optimalité que nous écrivons, nous cherchons à caractériser les variations locales de la fonction objectif autour d'une stratégie courante. Nous verrons qu'afin de se servir de cette information locale (le gradient), nous aurons besoin de discrétiser l'aléa ; nous introduirons alors des opérateurs d'interpolation-régression, comme cela avait été fait au §2.2 afin d'extraire une stratégie à partir de la valeur de la commande en des états particuliers. Nous ne rentrerons pas ici dans les détails de la méthode, qui sont développés dans l'article de Carpentier *et al.* (2009b).

Conditions d'optimalité

On revient au problème (1.3). Au §1.2.3 nous avons énoncé les conditions d'optimalité qui nous seront utiles pour la mise en œuvre de la méthode particulaire dite "espérance du gradient et espérance de l'état adjoint". Celles-ci ont la propriété de ne plus comporter d'espérances conditionnelles, mais seulement des espérances. On introduit tout d'abord le Lagrangien associé aux contraintes de dynamique sur $\boldsymbol{X}$:

$$\mathbb{E}\left(\sum_{t=0}^{T} C_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t\right) + \sum_{t=0}^{T-1} \boldsymbol{\lambda}_{t+1}^{\top}\left(f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right) - \boldsymbol{X}_{t+1}\right)\right),$$

avec $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_T \in L^2\left(\Omega, \mathcal{A}, \mathbb{P}\right)$ les multiplicateurs de Lagrange associés à la contrainte de dynamique de l'état. Ceux-ci ont par définition la même dimension que l'état : ils sont à valeurs dans $\mathbb{R}^n$. Il faut bien noter que ceux-ci sont des variables aléatoires. Comme il avait été discuté dans le chapitre 1, il est naturel d'introduire la version adaptée des multiplicateurs, qui s'écrit, à l'optimum : $\boldsymbol{\Lambda}_t := \mathbb{E}\left(\boldsymbol{\lambda}_t \middle| \boldsymbol{X}_t\right)$. Autrement dit, l'état adjoint est fonction de l'état du système. Pour simplifier l'écriture et parce qu'elles seront inutiles dans notre exemple, nous ne considérons pas ici de contraintes ponctuelles sur la commande. On reprend les conditions d'optimalité dans leur version la plus "intégrée", c'est-à-dire celle dans laquelle n'apparaissent que des espérances et pas d'espérances conditionnelles (voir la thèse de Dallagi, 2007, pour de plus amples détails). Il s'agit de conditions portant sur les variables de décision $\boldsymbol{X}$ et $\boldsymbol{U}$:

$$\boldsymbol{X}_0 = \boldsymbol{W}_0, \tag{2.10a}$$

$$\boldsymbol{X}_{t+1} = f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right), \tag{2.10b}$$

$$\boldsymbol{\Lambda}_T = \frac{\partial C_T}{\partial x}\left(\boldsymbol{X}_T, \boldsymbol{U}_T\right)^{\top}, \tag{2.10c}$$

$$\boldsymbol{\Lambda}_t = \frac{\partial C_t}{\partial x}\left(\boldsymbol{X}_t, \boldsymbol{U}_t\right)^{\top} + \mathbb{E}_{\boldsymbol{W}_{t+1}}\left(\frac{\partial f_t}{\partial x}\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right)^{\top} \boldsymbol{\Lambda}_{t+1}\right), \tag{2.10d}$$

$$0 = \frac{\partial C_t}{\partial u}\left(\boldsymbol{X}_t, \boldsymbol{U}_t\right)^{\top} + \mathbb{E}_{\boldsymbol{W}_{t+1}}\left(\frac{\partial f_t}{\partial u}\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right)^{\top} \boldsymbol{\Lambda}_{t+1}\right). \tag{2.10e}$$

Les deux espérances présentes dans les équations (2.10d) et (2.10e) portent uniquement sur le bruit $\boldsymbol{W}_{t+1}$. Rappelons que $\boldsymbol{\Lambda}_{t+1}$ est une variable aléatoire mesurable par rapport

8. Dans le travail de Dallagi (2007), on trouve en fait plusieurs versions de cette méthodologie rassemblées sous le nom de méthodes particulaires. Ici, si rien n'est précisé, nous utiliserons la version dite "espérance du gradient et espérance de l'état adjoint".

à $\boldsymbol{X}_{t+1} = f_t(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1})$. Nous n'avons donc pas ici d'espérances conditionnelles à calculer mais seulement des espérances. Les équations (2.10b) et (2.10a) représentent la dynamique de l'état, les équations (2.10c) et (2.10d) la dynamique (rétrograde en temps) de l'état adjoint, et l'équation (2.10e) indique que le gradient du Lagrangien est nul à l'optimum. Ce sont ces conditions d'optimalité que la méthode particulaire cherche à résoudre. Pour ce faire, on passe par une étape de discrétisation des espérances à l'aide d'une méthode de Monte-Carlo, qui aboutit à des conditions d'optimalité approchées.

Conditions d'optimalité approchées

Les conditions (2.10) sont difficiles à vérifier en l'état, notamment du fait des espérances. Les méthodes particulaires vont approcher ces conditions en utilisant des techniques de Monte-Carlo. Ainsi, on se donne N variables aléatoires $\boldsymbol{W}^1, \dots, \boldsymbol{W}^N$ i.i.d. de même loi que $\boldsymbol{W}$, que nous appelons un N-échantillon de $\boldsymbol{W}$. Cet échantillon nous sert à estimer les espérances présentes dans les équations (2.10d) et (2.10e). Étant donné cet échantillon, la méthode particulaire du type "espérance du gradient et espérance de l'état adjoint" cherche à résoudre les conditions suivantes :

$$\boldsymbol{X}_0^i = \boldsymbol{W}_0^i, \tag{2.11a}$$

$$\boldsymbol{X}_{t+1}^i = f_t\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^i\right), \tag{2.11b}$$

qui constituent la dynamique de l'état,

$$\boldsymbol{\Lambda}_T^i = \frac{\partial C_T}{\partial x}\left(\boldsymbol{X}_T^i, \boldsymbol{U}_T^i\right)^\top, \tag{2.11c}$$

$$\boldsymbol{\Lambda}_t^i = \frac{\partial C_t}{\partial x}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i\right)^\top + \frac{1}{N}\sum_{j=1}^N \left(\frac{\partial f_t}{\partial x}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j\right)^\top \tilde{\boldsymbol{\Lambda}}_{t+1}\left(f_t\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j\right)\right)\right), \tag{2.11d}$$

qui constituent la dynamique de l'état adjoint, et la condition de gradient nul :

$$0 = \frac{\partial C_t}{\partial u}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i\right)^\top + \frac{1}{N}\sum_{j=1}^N \left(\frac{\partial f_t}{\partial u}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j\right)^\top \tilde{\boldsymbol{\Lambda}}_{t+1}\left(f_t\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j\right)\right)\right), \tag{2.11e}$$

où $\tilde{\boldsymbol{\Lambda}}_{t+1}$ est un opérateur de régression utilisant les données $((\boldsymbol{X}_{t+1}^i)_i, (\boldsymbol{\Lambda}_{t+1}^i)_i)$. Pour bien comprendre l'intérêt de cet opérateur de régression, on peut partir du pas de temps final $t = T$ où, par l'équation (2.11c), on connaît les N états adjoints $\boldsymbol{\Lambda}_T^i$, chacun étant fonction de l'état $\boldsymbol{X}_T^i = f_{T-1}(\boldsymbol{X}_{T-1}^i, \boldsymbol{U}_{T-1}^i, \boldsymbol{W}_T^i)$. Afin de calculer l'état adjoint au pas de temps précédent (via l'équation d'évolution (2.11d) pour $t = T-1$) on a besoin de connaître la valeur de l'état adjoint à l'instant T pour les états $f_{T-1}(\boldsymbol{X}_{T-1}^i, \boldsymbol{U}_{T-1}^i, \boldsymbol{W}_T^j), \forall j = 1, \dots, N$. Cette collection d'états à l'instant T n'a aucune raison de faire partie de la liste de ceux en lesquels nous avons calculé les états adjoints $\boldsymbol{\Lambda}_T^i$. Nous approchons donc la valeur de l'état adjoint en ces états "mixtes" que l'on notera parfois $\boldsymbol{X}_T^{i,j}$ à l'aide d'un opérateur de régression basé sur les couples $((\boldsymbol{X}_T^i)_i, (\boldsymbol{\Lambda}_T^i)_i)$. À titre d'exemple de tels opérateurs, on peut citer le plus proche voisin qui est sans doute le plus simple et que nous utiliserons par la suite au §2.3.2 et au §2.3.3, ou bien les opérateurs à noyaux (Nadaraya, 1964, Watson, 1964, Devroye, 1987) qui consistent à faire des moyennes locales, pondérées par une fonction noyau (typiquement un noyau gaussien).

Nous venons de voir de quelle manière la méthode particulaire discrétise les conditions d'optimalité du problème de départ (1.3). Une fois ces conditions d'optimalité approchées résolues, nous obtenons une stratégie, fonction du nombre N de scénarios utilisés. Nous nous posons maintenant la même question que pour les scénarios : comment l'EQM associée à la stratégie particulaire, c'est-à-dire la stratégie vérifiant les conditions d'optimalité approchées (2.11), évolue-t-elle lorsque le nombre de scénarios augmente ? On montre au §2.3.2 les difficultés liées à l'étude de cette vitesse de convergence dans un cadre théorique général. On propose ensuite une expérimentation numérique, sur le même problème qu'au §2.2.3, qui permettra d'évaluer la vitesse de convergence sur un exemple et de la comparer à celle obtenue avec les arbres de scénarios.

2.3.2 Difficultés de l'analyse d'erreur dans le cas général

Si on revient à la condition (2.10e) qui impose la nullité du gradient à l'optimum, on observe que l'état adjoint $\boldsymbol{\Lambda}_{t+1}$ à l'instant $t+1$ est, à l'optimum, une fonction de l'état $\boldsymbol{X}_{t+1} = f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right)$. En utilisant le théorème des fonctions implicites, avec les hypothèses suffisantes, on pourra donc extraire une expression de $\boldsymbol{U}_t$ en fonction de l'état $\boldsymbol{X}_t$ et des caractéristiques statistiques du bruit $\boldsymbol{W}_{t+1}$.

Naturellement, on s'attend, en utilisant la condition (2.11e), à obtenir une expression de $\boldsymbol{U}_t^i$, pour chaque particule i. Nous allons expliciter ces conditions dans le cas où l'interpolateur choisi est le plus proche voisin. On rappelle la condition (2.11e) de nullité du gradient :

$$\frac{\partial C_t}{\partial u}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i\right)^\top + \frac{1}{N}\sum_{j=1}^{N}\left(\frac{\partial f_t}{\partial u}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j\right)^\top \tilde{\boldsymbol{\Lambda}}_{t+1}\left(f_t\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j\right)\right)\right) = 0.$$

On souhaiterait extraire de cette relation une expression de $\boldsymbol{U}_t^i$ comme fonction de $\boldsymbol{X}_t^i$. Notons $\zeta_{t+1}(i,j)$ l'indice de la particule d'état la plus proche (en norme L^2) de l'état à l'instant $t+1$, $f_t(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j)$. On a alors que $\tilde{\boldsymbol{\Lambda}}_{t+1}(f_t(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j)) = \boldsymbol{\Lambda}_{t+1}^{\zeta_{t+1}(i,j)}$, car l'interpolateur choisi est le plus proche voisin, ce qui donne la condition de nullité du gradient suivante :

$$\frac{\partial C_t}{\partial u}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i\right)^\top + \frac{1}{N}\sum_{j=1}^{N}\left(\frac{\partial f_t}{\partial u}\left(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j\right)^\top \boldsymbol{\Lambda}_{t+1}^{\zeta_{t+1}(i,j)}\right) = 0. \tag{2.12}$$

Or $\boldsymbol{\Lambda}_{t+1}^{\zeta_{t+1}(i,j)}$ dépend de $\boldsymbol{X}_{t+1}^{\zeta_{t+1}(i,j)}$, donc du triplet $(\boldsymbol{X}_t^{\zeta_{t+1}(i,j)}, \boldsymbol{U}_t^{\zeta_{t+1}(i,j)}, \boldsymbol{W}_{t+1}^{\zeta_{t+1}(i,j)})$, et non de $(\boldsymbol{X}_t^i, \boldsymbol{U}_t^i, \boldsymbol{W}_{t+1}^j)$. Deux remarques s'imposent alors :

1. le système d'équations n'est pas explicite : on ne peut pas tirer directement de l'équation (2.12) une expression de $\boldsymbol{U}_t^i$ en fonction de l'état $\boldsymbol{X}_t^i$, car on a un couplage entre les différentes particules de commande à l'instant t ;
2. l'application ζ dépend des particules de commande $\left(\boldsymbol{U}_t^i\right)_{i=1,\dots,N}$, ce qui rend le système non linéaire.

De ce fait, il sera difficile d'analyser l'erreur de manière analytique, même dans un cas particulier. Nous proposons donc maintenant une évaluation empirique de l'erreur commise.

2.3.3 Exemple d'un problème de filtrage

Nous reprenons le même exemple qu'au §2.2.3. Afin de le traiter par la méthode particulaire retenue, nous commençons par écrire les conditions d'optimalité du problème. Ainsi, à l'optimum :

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t + \boldsymbol{U}_t + \boldsymbol{W}_{t+1}, \qquad \forall t = 0, \dots, T-1, \tag{2.13a}$$
$$\boldsymbol{X}_0 = \boldsymbol{W}_0, \tag{2.13b}$$
$$\boldsymbol{\Lambda}_T = 2\boldsymbol{X}_T, \tag{2.13c}$$
$$\boldsymbol{\Lambda}_t = \mathbb{E}\left(\boldsymbol{\Lambda}_{t+1} \middle| \boldsymbol{X}_t\right), \qquad \forall t = 0, \dots, T-1, \tag{2.13d}$$
$$2\varepsilon\boldsymbol{U}_t + \mathbb{E}\left(\boldsymbol{\Lambda}_{t+1} \middle| \boldsymbol{X}_t\right) = 0, \qquad \forall t = 0, \dots, T-1. \tag{2.13e}$$

On peut noter qu'on retrouve facilement la relation (2.8) définissant la solution optimale du problème. Remarquons également que la relation (2.13e) peut s'écrire, à l'aide de l'équation (2.13d) :

$$2\varepsilon\boldsymbol{U}_t + \boldsymbol{\Lambda}_t = 0,$$

ce qui fait que, dans ce cas particulier, nous n'aurons à appliquer qu'une fois par pas de temps l'opérateur de régression.

Nous suivons alors la méthodologie décrite au §2.3.1 et discrétisons les conditions d'optimalité (2.13) à l'aide des N particules de bruit $\boldsymbol{W}^1, \dots, \boldsymbol{W}^N$, ce qui donne :

$$\boldsymbol{X}^i_{t+1} = \boldsymbol{X}^i_t + \boldsymbol{U}^i_t + \boldsymbol{W}^i_{t+1}, \qquad \forall t = 0, \dots, T-1, \tag{2.14a}$$
$$\boldsymbol{X}^i_0 = \boldsymbol{W}^i_0, \tag{2.14b}$$
$$\boldsymbol{\Lambda}^i_T = 2\boldsymbol{X}^i_T, \tag{2.14c}$$
$$\boldsymbol{\Lambda}^i_t = \frac{1}{N}\sum_{j=1}^{N} \tilde{\boldsymbol{\Lambda}}_{t+1}\left(\boldsymbol{X}^i_t + \boldsymbol{U}^i_t + \boldsymbol{W}^j_{t+1}\right), \qquad \forall t = 0, \dots, T-1, \tag{2.14d}$$
$$2\varepsilon\boldsymbol{U}^i_t + \frac{1}{N}\sum_{j=1}^{N} \tilde{\boldsymbol{\Lambda}}_{t+1}\left(\boldsymbol{X}^i_t + \boldsymbol{U}^i_t + \boldsymbol{W}^j_{t+1}\right) = 0, \qquad \forall t = 0, \dots, T-1. \tag{2.14e}$$

Il nous faut maintenant résoudre les conditions (2.14). Conformément à l'esprit des méthodes particulaires, on utilise la relation (2.11e) afin de mettre en œuvre un algorithme de gradient avec recherche linéaire.

Une fois l'algorithme arrêté, nous obtenons une stratégie particulaire $\boldsymbol{\Gamma}^\sharp$, pour reprendre les notations du §2.1. Nous répétons cette expérience un grand nombre de fois[9] afin de calculer le biais et la variance associés à la stratégie particulaire. À l'image de ce qui avait été effectué pour les arbres de scénarios, on trace dans la figure 2.4 les différentes stratégies impliquées dans le calcul du biais et de la variance au premier pas de temps.

Dans l'encart de gauche, on observe la stratégie optimale au premier pas de temps γ_0^* et la stratégie approchée moyenne $\gamma_0^\sharp$ obtenue par méthode particulaire. On a utilisé pour ce faire 81 scénarios, comme cela avait été le cas dans la figure 2.2. Dans le cas des arbres, avec un taux de branchement régulier et pour ce problème à 4 pas de temps, l'utilisation de 81 scénarios impliquait que l'arbre était constitué de seulement 3 nœuds au premier pas de temps, $3^2 = 9$ au deuxième, etc. Avec la méthode particulaire, le nombre de nœuds est le même à chaque pas de temps. Il n'est donc pas étonnant de constater que la distance

9. ici 10^5 fois, ce qui nous paraît suffisant pour négliger l'erreur commise lors de l'estimation des espérances de (2.3) par Monte-Carlo

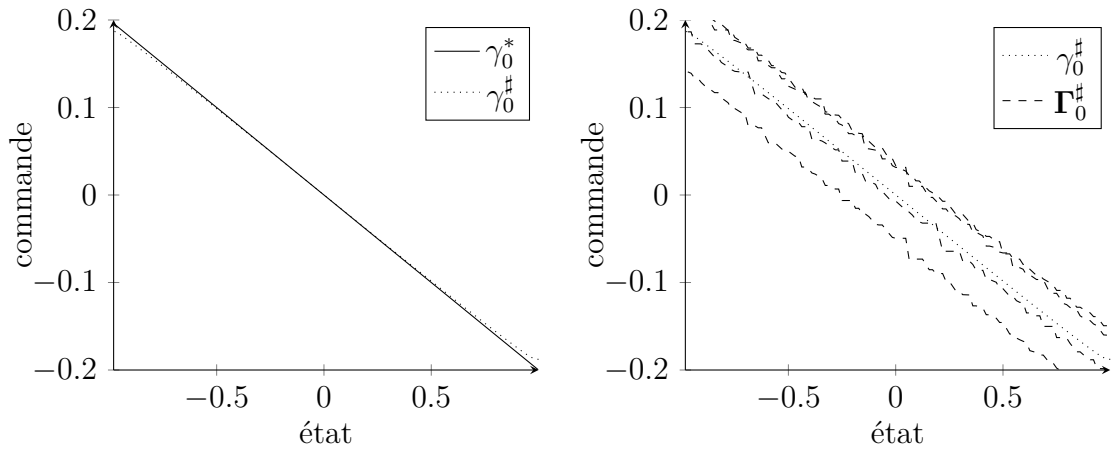

FIGURE 2.4 – *Feedbacks* exacts et approchés issus de la méthode particulaire.

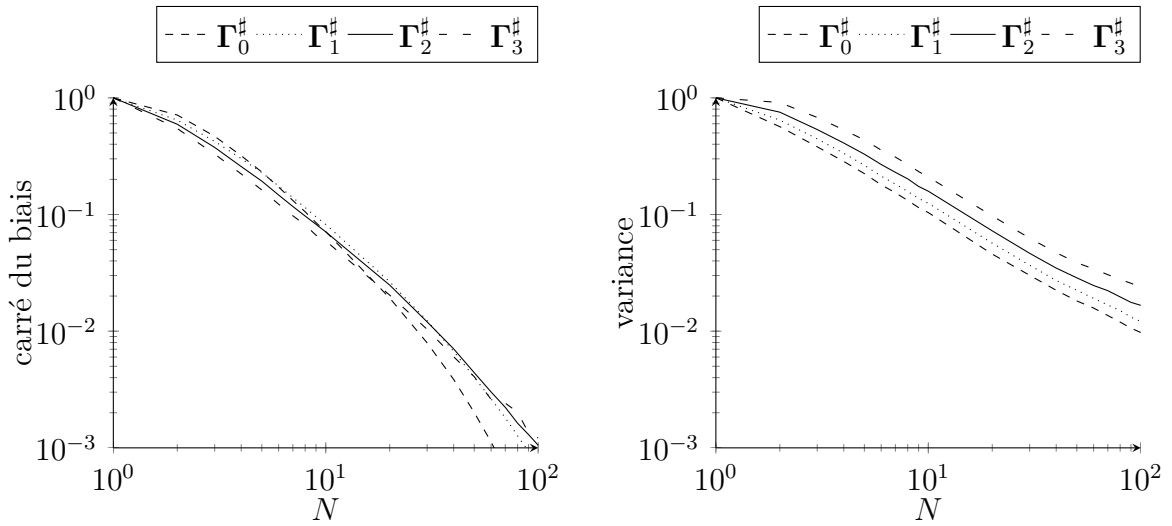

FIGURE 2.5 – Carré du biais et variance de la commande par méthode particulaire en fonction du nombre de particules.

entre les stratégies optimale et approchée moyenne (qui n'est autre que le biais) est très faible par rapport à celle que l'on observait dans la figure 2.2.

Dans l'encart de droite, on observe la stratégie approchée moyenne $\gamma^\sharp$, ainsi que plusieurs stratégies approchées particulières. Même s'il est difficile de les distinguer, on a bien affaire à des fonctions constantes par morceaux, ces derniers étant au nombre de 81. Rappelons que le terme de variance de l'EQM est la distance moyenne entre les stratégies approchées (les tirages de $\boldsymbol{\Gamma}^\sharp$) et la stratégie approchée moyenne $\gamma^\sharp$.

On présente maintenant l'évolution du carré du biais et de la variance en fonction du nombre N de particules choisi. Les résultats pour un horizon de temps $T = 4$ (donc 4 stratégies) sont présentés dans la figure 2.5.

Dans chaque graphique nous observons quatre courbes, correspondant aux stratégies associées aux quatre pas de temps du problème. Pour un nombre de particules N variant de 1 à 100, on a tracé sur une échelle logarithmique l'évolution des quatre carrés des biais (dans l'encart de gauche) et des quatre variances (dans l'encart de droite).

On observe que les biais associés aux stratégies particulaires varient à la même vitesse quel que soit le pas de temps. De plus, les courbes semblent connaître une asymptote affine (dans l'échelle logarithmique) : il apparaît ainsi que le carré du biais décroît au moins comme N^{-2} au voisinage de 100 particules. Par ailleurs, la variance décroît clairement en N^{-1} quel que soit le pas de temps considéré. Ainsi le carré du biais reste toujours négligeable devant la variance, pour laquelle on ne pourrait espérer une convergence plus rapide. On retrouve en effet le taux de convergence des méthodes de Monte-Carlo. Pour résumer, sur cet exemple, l'EQM associée à la méthode particulaire décroît comme N^{-1}, N étant le nombre de scénarios utilisés.

À ce stade il est important de se souvenir que le taux de décroissance de l'EQM observé pour les arbres de scénarios sur le même exemple était de l'ordre de n_b^{-1}, n_b étant le taux de branchement de l'arbre. Autrement dit, l'EQM associée à la discrétisation par arbres de scénarios varie comme $N^{-1/T}$, ce qui, d'une part, est inférieur à ce que nous observons pour les méthodes particulaires et, d'autre part, croît de manière exponentielle en fonction de l'horizon de temps T du problème. Ainsi, pour obtenir la même précision qu'une méthode particulaire utilisant N scénarios, il faudrait, avec une méthode d'arbres de scénarios, utiliser un taux de branchement de l'ordre de N. On obtiendrait ainsi un arbre contenant N^T nœuds !

Finalement, on conclut que le nombre de scénarios nécessaire lors de la discrétisation à l'aide des méthodes particulaires :

- ne varie pas en fonction de l'horizon de temps du problème ;
- est nettement inférieur à celui requis par les arbres de scénarios, qui lui varie exponentiellement avec l'horizon de temps.

Il faut remarquer que, dans l'exemple considéré ici, la dimension de l'espace d'état est de 1. Or on sait bien qu'une difficulté commune à toutes les méthodes numériques cherchant à résoudre des problèmes de commande optimale stochastique est la dimension de l'espace d'état. À titre d'exemple, la programmation dynamique, très efficace en petite dimension, n'est pas utilisable en pratique pour une dimension de l'espace d'état supérieure à 5, environ. Autrement dit, même s'il semble clair que l'approche particulaire a, grâce à la manière dont elle discrétise la structure probabiliste du problème, un meilleur taux de convergence que les méthodes d'arbres de scénarios pour les problèmes à plusieurs pas de temps, il convient d'étudier un exemple en plus grande dimension d'espace d'état afin d'observer comment chacune de ces méthodes se comporte.

2.4 Question de la dimension

Les résultats présentés au §2.2.3 et au §2.3.3 se rapportent à un problème dont la dimension de l'espace d'état est de 1. Soit N le nombre de scénarios utilisés dans l'une ou l'autre des discrétisations. Les résultats numériques montrent que, si l'erreur liée à la discrétisation par méthode particulaire varie comme N^{-1}, celle liée à la discrétisation par arbres de scénarios varie comme $N^{-\frac{1}{T}}$, où T est l'horizon de temps du problème. Autrement dit, l'erreur liée aux arbres de scénarios varie comme l'inverse du taux de branchement et il faudra donc, pour obtenir une précision donnée, utiliser un nombre de scénarios variant de manière exponentielle en fonction de l'horizon de temps T du problème.

Le fait d'avoir travaillé sur un exemple numérique, à défaut d'avoir pu mener à bien l'étude théorique générale de vitesse de convergence de ces méthodes, n'allège pas le discours concernant les méthodes arborescentes. En effet, on ne peut espérer obtenir une

vitesse de convergence dans le cas général meilleure que dans un cas particulier. Le fait que le nombre de scénarios nécessaires à l'obtention d'une précision donnée varie de façon exponentielle avec l'horizon de temps est un défaut majeur lorsque l'on souhaite résoudre des problèmes à plusieurs pas de temps, comme c'est notre cas ici. Pour cette raison, il est donc clair que les méthodes particulaires sont un meilleur candidat pour la résolution numérique de problèmes de commande optimale stochastique à plusieurs pas de temps.

Cependant, afin d'étayer la comparaison des deux méthodologies de discrétisation, que l'on peut présenter comme "discrétisation-optimisation" pour les arbres de scénarios à l'inverse de "optimisation-discrétisation" pour les méthodes particulaires, il convient de se poser la question de l'influence de la dimension de l'espace d'état. En effet, il s'agit d'une des difficultés majeures que rencontrent les méthodes numériques s'attaquant aux problèmes de commande optimale stochastique. L'exemple canonique est celui de la programmation dynamique (Bellman, 1957, Bertsekas, 2000). Cette méthode, que l'on a présenté au §1.2.1, souffre de la *malédiction de la dimension* : sa complexité varie de manière exponentielle en fonction de la dimension de l'espace d'état ! Ceci est une barrière que même l'amélioration considérable des moyens de calculs n'a pas réussi à repousser au-delà de petites dimensions.

Nous montrons maintenant que, bien qu'efficaces pour des problèmes de petite dimension d'état, les méthodes à base de scénarios que nous étudions ici ne sont pas une réponse à la malédiction de la dimension. Cela justifie, pour les problèmes de grande taille, l'investissement dans des méthodes de décomposition.

Formulation du problème

Pour ces raisons, nous introduisons un problème de dimension d, sur lequel nous déclinerons les deux méthodologies de discrétisation présentées ici pour $d = 2$. Nous ne rappellerons pas la mise en œuvre des deux méthodes sur ce problème, qui ne diffère pas de ce qui a été présenté dans l'exemple précédent. L'exemple considéré est une extension directe du problème de filtrage précédent à une dimension quelconque (finie) d'état. On choisit de conserver une commande de dimension 1 afin de faciliter l'étude des résultats. Ce choix n'a pas d'influence sur notre étude.

Soient $\boldsymbol{W}_0, \dots, \boldsymbol{W}_T$ des variables aléatoires i.i.d. de loi uniforme sur $[-1, 1]^d$. On considère le problème de commande optimale stochastique :

$$\min_{(\boldsymbol{U}_0,\dots,\boldsymbol{U}_{T-1})} \quad \mathbb{E}\left(\varepsilon \sum_{t=0}^{T-1} \boldsymbol{U}_t^2 + \boldsymbol{X}_T^\top \boldsymbol{X}_T\right), \tag{2.15a}$$

$$\text{s.c.} \quad \boldsymbol{X}_{t+1}^i = \boldsymbol{X}_t^i + \boldsymbol{U}_t + \boldsymbol{W}_{t+1}^i, \quad \forall i = 1, \dots, d, \forall t = 0, \dots, T-1, \tag{2.15b}$$

$$\boldsymbol{X}_0 = \boldsymbol{W}_0, \tag{2.15c}$$

$$\boldsymbol{U}_t \preceq \boldsymbol{W}_0, \dots, \boldsymbol{W}_t. \tag{2.15d}$$

De la même manière que précédemment, du fait de l'indépendance temporelle des bruits, on sait que l'on peut chercher la stratégie optimale en *feedback* sur l'état x. Nous comparons ici différentes stratégies entre elles.

Résolution numérique par arbres de scénarios

On commence par traiter le problème par méthode d'arbres de scénarios, toujours avec un taux de branchement régulier. On effectue donc n_b tirages de $\boldsymbol{W}_0$, n_b^2 tirages de $\boldsymbol{W}_1$, …, n_b^{T+1} tirages de $\boldsymbol{W}_T$. Une fois ces tirages effectués, on réécrit le problème (2.15) sur

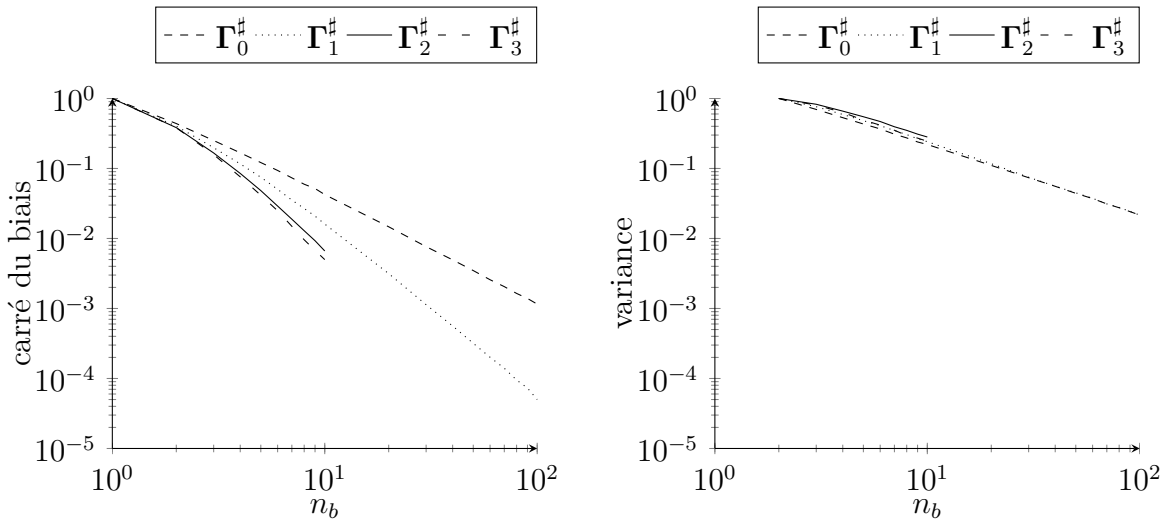

FIGURE 2.6 – Carré du biais et variance de la commande par arbres de scénarios en fonction du taux de branchement, pour un état de dimension 2.

l'arbre. On sait résoudre ce problème linéaire quadratique de manière analytique, comme cela avait été fait au §2.2.3. Toujours comme précédemment, on "étend" les commandes obtenues en les nœuds de l'arbre, à l'aide d'un opérateur de régression [10]. On répète cette expérience un grand nombre de fois (ici 10^4) afin d'estimer de manière satisfaisante les espérances présentes dans le calcul de l'EQM de la relation (2.3) : celle qui apparaît dans le terme de variance et celle (cachée) qui apparaît dans le calcul de la stratégie approchée moyenne $\gamma^\sharp$. Ainsi, pour chaque pas de temps, on trace dans la figure 2.6 l'évolution du carré du biais (encart de gauche) et de la variance (encart de droite), tels qu'ils sont définis dans la relation (2.3), en fonction du taux de branchement n_b de l'arbre.

On observe alors que le carré du biais est, comme c'était le cas en dimension 1 d'état, négligeable devant la variance. Cependant, il faut noter qu'il décroît de manière beaucoup plus lente qu'auparavant. On s'attendait naturellement à ce comportement : à n_b fixé, soit avec le même nombre de points que dans le cas de l'espace d'état de dimension 1, on cherche à représenter une fonction (la stratégie) dont l'espace de départ est maintenant de dimension 2. On observe le même phénomène sur la variance : elle décroît légèrement plus lentement qu'auparavant. Il faut observer que le terme de variance n'est pas seulement influencé par un phénomène du type "Monte-Carlo" : l'erreur due à la régression intervient également dans le terme de variance de l'équation (2.3), via les stratégies $\mathbf{\Gamma}^\sharp$ et $\gamma^\sharp$. Il n'est donc pas surprenant que la variance décroisse plus lentement que n_b^{-1}, qui est la vitesse associée aux estimateurs de Monte-Carlo pour le calcul numérique de l'espérance.

Résolution numérique par méthode particulaire

De la même manière, on trace dans la figure 2.7 les résultats obtenus par méthode particulaire pour la résolution du problème (2.15). Dans l'encart de gauche on a représenté l'évolution du carré du biais, tel qu'il est défini dans la relation (2.3) en fonction du nombre de particules N utilisées. Dans l'encart de droite, on a représenté l'évolution de la

10. L'opérateur choisi ici est le plus proche voisin. Pour chaque valeur d'état où l'on souhaite calculer la commande, on commence par rechercher parmi les nœuds de l'arbre la valeur d'état la plus proche au sens de la distance euclidienne dans $\mathbb{R}^2$, et on applique la décision correspondante.

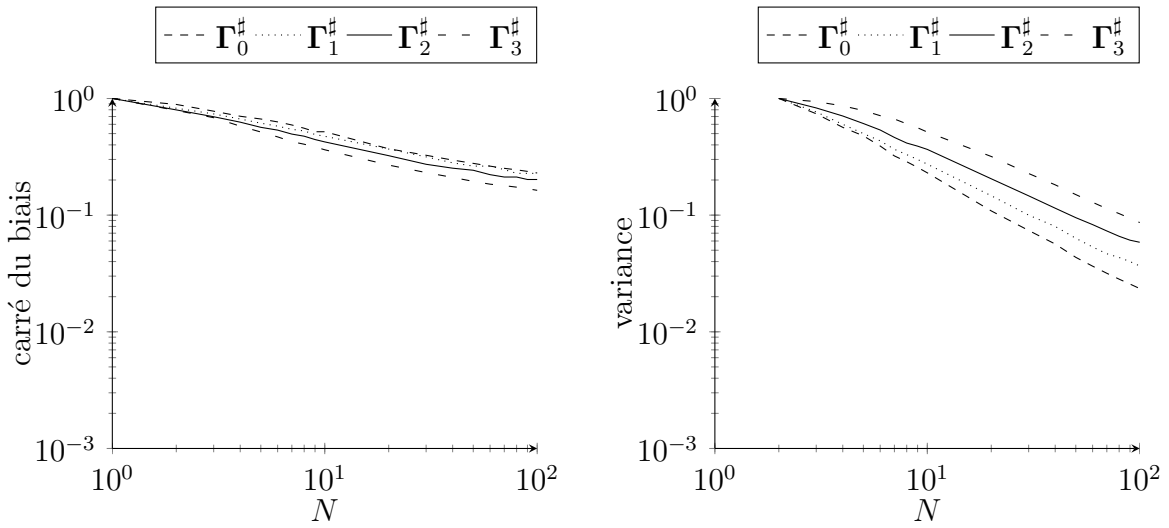

FIGURE 2.7 – Carré du biais et variance de la commande par méthode particulaire en fonction du nombre de particules, pour un état de dimension 2.

variance, telle qu'elle est définie dans la même relation, toujours en fonction du nombre de particules.

On observe une nette diminution de la vitesse de convergence du biais par rapport au cas de l'espace d'état de dimension 1 considéré précédemment. Le carré du biais décroît maintenant comme $N^{-0.5}$, contre $N^{-1.5}$ dans le cas de la dimension 1. De plus, on observe que celui-ci devient maintenant prépondérant devant la variance. Nous nous attendions naturellement à ce que l'erreur du schéma décroisse plus lentement lorsque la dimension augmente. Cela dit, on peut espérer repousser ce point à une dimension supérieure en choisissant un autre opérateur de régression que le plus proche voisin, qui est le plus frustre d'entre eux. Il faut cependant rester conscient qu'à partir d'une certaine dimension, la méthode particulaire demandera d'utiliser un trop grand nombre de particules afin d'obtenir une erreur raisonnable, ce quel que soit l'opérateur de régression utilisé. En ce sens, elle ne répond pas à la malédiction de la dimension, pas plus que les méthodes d'arbres de scénarios.

Cette vitesse de convergence est très spécifique au problème étudié et à la structure de sa solution. Les méthodes particulaires ont en effet, comme les méthodes d'arbres de scénarios, la propriété de concentrer les particules d'état et de commande dans les régions "intéressantes", c'est-à-dire le support des lois de l'état optimal et de la commande optimale. Ainsi, plus ces supports seront petits, plus les méthodes présentées ici seront performantes, même dans des espaces d'état de grande dimension.

Lors de la comparaison des graphiques de vitesse de convergence des arbres de scénarios et de la méthode particulaire, il faut bien garder à l'esprit la nature différente des abscisses. Ainsi les graphiques concernant les arbres de scénarios sont-ils tracés en fonction du taux de branchement n_b, alors que ceux concernant les méthodes particulaires sont tracés en fonction du nombre de particules (scénarios). Plus précisément, on observe dans la figure 2.6 que pour diviser l'EQM par 10, il convient de multiplier par 10 le taux de branchement de l'arbre. Pour la méthode particulaire, on observe dans la figure 2.7 que pour diviser l'EQM par 10, il convient de multiplier le nombre de particules par 100. Or, pour obtenir un taux de branchement de 10 pour un arbre de scénarios dans notre

problème à 4 pas de temps, il est nécessaire de tirer 10^4 scénarios, contre seulement 100 scénarios pour les méthodes particulaires.

Conclusion

Nous avons mis en évidence la manière dont la vitesse de convergence de deux schémas de discrétisation de problèmes de commande optimale stochastique évolue en fonction du nombre de scénarios intervenant dans cette discrétisation et de la dimension de l'espace d'état du problème. On observe que les méthodes à base de scénarios étudiées ici souffrent d'une certaine forme de malédiction de la dimension : l'erreur du schéma se dégrade lorsque la dimension de l'espace d'état augmente et le nombre de scénarios nécessaire à l'obtention d'une précision donnée augmente parfois de façon importante. L'étude d'un problème de dimension supérieure à 2 n'apporterait pas d'éléments supplémentaires, notamment car ces propriétés dépendent fortement de la structure du problème particulier considéré. Il pourrait cependant être intéressant d'observer les comportements asymptotiques en la dimension de l'espace d'état, mais cela ne semble pas possible avec le protocole expérimental défini ici.

En conclusion, la vitesse de convergence des deux schémas diminue de manière importante lorsque la dimension de l'espace d'état augmente. La méthode particulaire, qui a un meilleur comportement que la méthode d'arbre de scénarios vis-à-vis du nombre de pas de temps du problème, n'est pas une réponse à la malédiction de la dimension. Son bon comportement en petite dimension en fait en revanche une bonne candidate dans le contexte d'une méthode de décomposition, pour la résolution des sous-problèmes où la dimension de l'état est de petite taille. Nous étudions de telles méthodes dans le chapitre 3.

2.5 Lien entre méthode particulaire et arbres de scénarios

Dans cette section, on mène une réflexion quelque peu parallèle au sujet traité dans ce chapitre. On espère éclairer le lecteur sur la raison pour laquelle la méthodologie "particulaire" offre de meilleurs résultats que la méthodologie "arborescente". D'une certaine manière, la méthode particulaire tire plus d'informations à partir des scénarios que la programmation stochastique et a donc besoin de moins de scénarios pour obtenir la même précision que sa concurrente.

On a pu constater, à la lecture des deux précédentes sections, des traits communs à la fois aux arbres de scénarios et aux méthodes particulaires, notamment l'utilisation de scénarios. On tente ici d'affiner la comparaison entre ces deux discrétisations. Ainsi, on se demande si les conditions (2.11) proviennent elles-mêmes d'un problème d'optimisation (voir figure 2.8). Nous montrons qu'en fait les conditions (2.11) peuvent être obtenues à partir des conditions d'optimalité du problème que l'on aurait discrétisées sur un arbre de scénarios.

Nous repartons du problème (2.1) et discrétisons directement l'aléa à l'aide des échantillons $\boldsymbol{W}^1, \dots, \boldsymbol{W}^N$. On obtient ainsi un problème d'optimisation écrit sur un arbre, dont nous décrirons les conditions d'optimalité. Les bruits étant indépendants, on peut se servir de ces N échantillons pour construire un arbre à N^T feuilles. En effet, l'aléa $\boldsymbol{W}_{t+1}$ est indépendant de $\boldsymbol{W}_t^i$, quel que soit $i = 1, \dots, N$. On peut donc se servir de tous les échantillons $\boldsymbol{W}_{t+1}^1, \dots, \boldsymbol{W}_{t+1}^N$ pour la représenter, et ramifier N fois pour chaque noeud

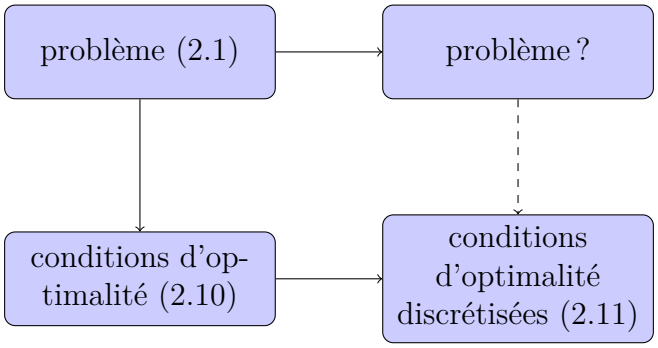

FIGURE 2.8 – Relations entre le problème, les conditions d'optimalité, et les conditions d'optimalité discrétisées.

de l'arbre, ce qui nous mène à N^T feuilles.

Afin d'écrire un problème sur un arbre de scénarios, on a introduit au §2.2.1 une fonction père ν :

$$x = \nu(y) \iff x \text{ est le père de } y,$$

$$y \in \nu^{-1}(x) \iff y \text{ est un des fils de } x.$$

De plus, on indexera ici l'état, la commande, ou le bruit à l'aide de l'indice du nœud, et non en fonction du temps. Enfin, le pas de temps du noeud i est noté $\theta(i)$. Le problème sur l'arbre s'écrit alors [11] :

$$\min_{x,u} \quad \sum_{i\in\mathcal{N}} \pi(i)\cdot C_{t(i)}\left(x_i, u_i\right), \tag{2.16a}$$

$$\text{s.c.} \quad x_i = f_{t(\nu(i))}\left(x_{\nu(i)}, u_{\nu(i)}, w_i\right), \qquad \forall i \in \mathcal{N}\backslash\{0\}, \tag{2.16b}$$

$$x_0 = w_0. \tag{2.16c}$$

On s'est ainsi ramené à un problème déterministe dans lequel on cherche à minimiser la moyenne des coûts sur toutes les combinaisons d'aléas possibles (on rappelle que les bruits sont indépendants pas de temps par pas de temps). On peut observer que la contrainte de non-anticipativité qui était présente dans le problème (2.1) a disparu de cette formulation et est en quelque sorte codée "en dur" dans la structure d'arbre.

Écrivons maintenant les conditions d'optimalité du problème (2.16). On a la dynamique en avant de l'état :

$$x_i = f_{t(\nu(i))}\left(x_{\nu(i)}, u_{\nu(i)}, w_i\right), \quad \forall \nu \neq 0, \tag{2.17a}$$

$$x_0 = w_0, \tag{2.17b}$$

la dynamique en arrière de l'état adjoint :

$$\Lambda_i = \frac{\partial C_T}{\partial x}\left(x_i, u_i\right), \quad \forall i \in \theta^{-1}(T), \tag{2.17c}$$

$$\Lambda_i = \frac{\partial C_t}{\partial x}\left(x_i, u_i\right) + \frac{1}{N}\sum_{j\in\nu^{-1}(i)}\left(\Lambda_j^\top \frac{\partial f_t}{\partial x}\left(x_i, u_i, w_j\right)\right), \quad \forall i \in \mathcal{N}\backslash\theta^{-1}(T), \tag{2.17d}$$

11. Encore une fois, afin de simplifier l'exposé, on ne considère pas de contraintes ponctuelles sur la commande. Cela n'a pas d'influence sur le fond du discours.

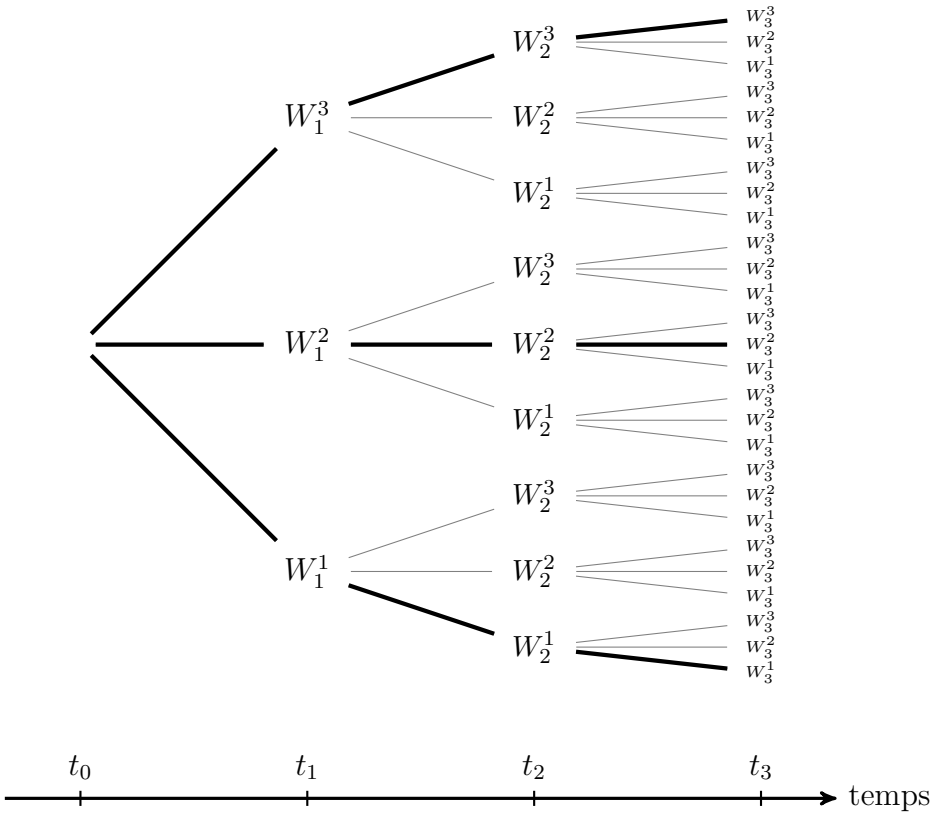

FIGURE 2.9 – Lien entre arbres de scénarios et méthode particulaire.

et la condition de nullité du gradient :

$$0 = \frac{\partial C_t}{\partial u}(x_i, u_i) + \sum_{j \in \nu^{-1}(i)} \left(\Lambda_j^\top \frac{\partial f_t}{\partial u}(x_i, u_i, w_j) \right), \quad \forall i \in \mathcal{N}. \tag{2.17e}$$

Les conditions (2.17) introduisent nettement un couplage entre chaque noeud et l'ensemble de ses fils, rendant impossible une réécriture de ces conditions sur un sous-arbre. Prenons par exemple la mise à jour de l'état adjoint du nœud i, donnée par (2.17d). Elle nécessite l'évaluation de l'état adjoint en tous les fils de i, soit sur l'ensemble $\nu^{-1}(i)$.

Pour détruire ce couplage, on peut choisir de remplacer les valeurs par une approximation de ces valeurs n'utilisant qu'un sous-ensemble des nœuds. Ainsi, considérons parmi les N^T feuilles de l'arbre les N feuilles correspondant aux échantillons de départ $\boldsymbol{W}^1, \dots, \boldsymbol{W}^N$, et notons Θ_T l'ensemble de ces N noeuds terminaux. Introduisons alors les sous-ensembles de noeuds :

$$\Theta_t = \{\nu(i), \forall i \in \Theta_{t+1}\}, \quad \forall t = 0, \dots, T-1,$$

et $\Theta = \cup_{t=0}^T \Theta_t$, qui est donc un sous-arbre à N feuilles, ne ramifiant qu'une seule fois : à l'origine. Ainsi, on représente dans la figure 2.9 l'arbre de scénarios construit à partir des N scénarios $\boldsymbol{W}^1, \dots, \boldsymbol{W}^N$ que l'on a "brassés" grâce à l'hypothèse d'indépendance des bruits, afin de construire l'arbre complet à N^T feuilles. On voit ici clairement l'explosion du nombre de nœuds avec l'horizon de temps. Au contraire, la méthode particulaire ne se sert que des scénarios "historiques" $\boldsymbol{W}^1, \dots, \boldsymbol{W}^N$, que l'on a représenté par des traits plus

épais. Lorsqu'on a besoin de calculer des espérances conditionnelles (par exemple dans la condition d'optimalité (2.10e)), on se sert de l'information calculée sur les particules, que l'on extrapole au point qui nous intéresse via un opérateur d'interpolation/régression (voir équation (2.11e)).

On réécrit alors les conditions (2.17) sur le sous-arbre Θ, en s'efforçant de n'utiliser que les noeuds de celui-ci. On a de la même façon la dynamique de l'état :

$$x_i = f_{t(\nu(i))}\left(x_{\nu(i)}, u_{\nu(i)}, w_i\right), \quad \forall i \in \Theta\backslash\{0\}, \tag{2.18a}$$

$$x_0 = w_0, \tag{2.18b}$$

la dynamique "approchée" de l'état adjoint :

$$\Lambda_i = \frac{\partial C_T}{\partial x}(x_i, u_i), \quad \forall i \in \Theta_T, \tag{2.18c}$$

$$\Lambda_i = \frac{\partial C_t}{\partial x}(x_i, u_i) + \frac{1}{N}\sum_{j\in\nu^{-1}(i)}\left(\tilde{\Lambda}_j^\top \frac{\partial f_t}{\partial x}(x_i, u_i, w_j)\right), \quad \forall i \in \Theta\backslash\Theta_T, \tag{2.18d}$$

et la condition "approchée" de nullité du gradient :

$$0 = \frac{\partial C_t}{\partial u}(x_i, u_i, w_j) + \sum_{j\in\nu^{-1}(i)}\left(\tilde{\Lambda}_j^\top \frac{\partial f_t}{\partial u}(x_i, u_i, w_j)\right), \quad \forall i \in \Theta, \tag{2.18e}$$

où $\tilde{\Lambda}_j = \Lambda_j$ si $j \in \Theta$ et opère une régression à partir de $(\Lambda_j)_{j\in\Theta}$ sinon.

On remarque alors que les conditions (2.18) sont strictement les mêmes que les conditions (2.11). On peut donc voir la méthode particulaire comme une technique réduisant drastiquement le nombre de branches d'un arbre de scénarios en utilisant des opérateurs de régression.

2.6 Conclusion

La résolution numérique de problèmes de commande optimale stochastique passe couramment par la discrétisation de la structure d'information sous la forme d'un arbre de scénarios, censé représenter la "diffusion" des aléas avec le temps. Cette méthodologie a connu un franc succès du fait de sa simplicité et parce qu'elle permet de se ramener à un problème déterministe pour lequel on saura utiliser des méthodes éprouvées de l'optimisation déterministe (programmation linéaire, algorithmes variationnels, décomposition, etc.).

Cependant, nous montrons dans cette partie que les arbres de scénarios souffrent d'un défaut important : le nombre de nœuds dans l'arbre nécessaire à l'obtention d'une précision donnée varie de manière exponentielle avec l'horizon de temps, ce qui rend cette technique inefficace pour des problèmes comprenant un grand nombre de pas de temps. Ce défaut provient de la structure même de l'arbre : afin de calculer avec suffisamment de précision les espérances, il convient de fixer un taux de branchement n_b suffisant en chaque nœud de l'arbre. Mais cela implique que le nombre de nœuds dans l'arbre sera de l'ordre de n_b^T, si on note T l'horizon de temps.

L'algorithme particulaire que nous utilisons demande un nombre de calculs croissant comme le cube du nombre de scénarios utilisés, ce qui rend difficile, sur les exemples

considérés, la mise en œuvre de la méthode avec plus de 100 scénarios environ. Or cela est peu pour une méthode de Monte-Carlo. Notons que nous n'avons pas travaillé ici à l'accélération de la méthode, qui consiste pour le moment à effectuer un algorithme de gradient avec recherche linéaire pour résoudre les conditions d'optimalité du problème. Dans le cadre classique de l'optimisation convexe, on sait qu'il existe un grand nombre de méthodes bien plus performantes que le gradient avec recherche linéaire. Ainsi reste-t-il certainement des travaux à mener autour de l'adaptation de ces méthodes évoluées à la résolution des conditions d'optimalité approchées décrites par les méthode particulaires. Cela permettrait de diminuer cette complexité et de pouvoir augmenter le nombre de scénarios utilisés et, par là même, la qualité des stratégies solutions.

Nous procédons ensuite à la même étude d'erreur sur la méthode particulaire proposée par Dallagi (2007) et observons que celle-ci ne souffre pas du même défaut. Sur l'exemple considéré, on montre que l'on est capable d'obtenir une précision bien supérieure à celle obtenue par arbres de scénarios, à nombre de scénarios égal. En revanche, cette technique de résolution ne fournit pas une réponse à la malédiction de la dimension : on observe sur un exemple où la dimension de l'espace d'état est de 2 que la vitesse de convergence diminue de manière importante par rapport au cas en dimension 1. Il faut noter que ce constat est très dépendant de la structure du problème étudié : puisque les particules ont tendance à se concentrer dans la région support de la loi de l'état optimal, on aura de bien meilleurs résultats dans les cas où l'état optimal est peu dispersé que dans les cas où il visite tout l'espace.

Nous nous sommes placés ici dans un cadre markovien : nous avons fait l'hypothèse que les bruits étaient indépendants d'un pas de temps à l'autre. Lorsque cela n'est pas le cas, on peut toujours s'y ramener quitte à intégrer dans la variable d'état les statistiques nécessaires à la vérification de l'hypothèse d'indépendance des bruits. Au plus, on devra intégrer dans l'état tout le passé des bruits afin de retrouver une propriété de Markov. Mais, dès lors, nous serons confrontés au problème de la dimension de l'état cité plus haut. Ces conclusions nous incitent à étudier l'application des méthodes de décomposition aux problèmes qui nous intéressent, ce que nous faisons dans le chapitre suivant.

Chapitre 3

Décomposition de problèmes de commande optimale stochastique

> Trop de connaissance ne facilite pas
> les plus simples décisions.
>
> *Les enfants de Dune*
> Frank Herbert (1920-1986)

La résolution de problèmes de commande optimale stochastique consiste à rechercher des stratégies de commande d'un système dynamique influencé par des bruits exogènes afin de minimiser un certain critère. Comme nous avons pu le voir au cours des chapitres précédents, les méthodes numériques de résolution de tels problèmes (que ce soit les méthodes classiques telles que la programmation stochastique et la programmation dynamique, ou des méthodes plus récentes telles que les méthodes particulaires) rencontrent toutes certaines difficultés lorsque le nombre de variables caractérisant le système est grand.

Dans le cadre déterministe, les méthodes de décomposition permettent de remplacer la résolution d'un problème de grande taille par des résolutions successives de problèmes de plus petite taille tout en garantissant, sous certaines hypothèses, que la commande ainsi trouvée correspond bien à la commande optimale du système de départ. Il existe un grand nombre de tels algorithmes de décomposition qui ont été rassemblés via le Principe du Problème Auxiliaire (PPA) par Cohen (1978). Dans le cas de problèmes stochastiques en boucle ouverte, c'est-à-dire lorsque l'on s'interdit de faire dépendre les décisions à un certain instant des observations passées du système, cette approche a été mariée avec l'algorithme du gradient stochastique dans la thèse de Culioli (1987) (voir aussi l'article de Cohen et Culioli, 1990).

Dans le cadre de la boucle fermée, la problématique est sensiblement différente. En effet, il est généralement profitable de se servir, à un instant donné, des observations passées faites sur le système afin d'adapter la décision courante. Dans ce contexte, nous sommes donc à la recherche de stratégies plutôt que de simples décisions : ce sont des fonctions qui, à chaque instant et à tout historique possible du système, associent une décision à prendre. La difficulté que cela pose aux algorithmes classiques de décomposition est alors assez simple à énoncer : la complexité du problème ne provient plus seulement de la taille de l'espace de décision mais également de la taille de l'espace d'observation. Or ce dernier ne se trouve pas réduit à l'issue d'une approche de décomposition "standard".

Ainsi, observant que l'information qui sert de base à la décision est de trop grande taille, plusieurs méthodes numériques ont été proposées pour la diminuer en espérant garder une décision "de qualité raisonnable". Prenons l'exemple d'un ensemble d'unités de production d'énergie avec chacune ses propres contraintes physiques de fonctionnement et ses propres aléas. Si les unités n'ont pas d'influence les unes sur les autres, alors il est naturel de rechercher des stratégies de gestion qui, pour chaque unité, ne dépendent que de l'information "locale" au sous-système, soit le passé des bruits locaux. Ajoutons maintenant une simple contrainte scalaire à chaque pas de temps, demandant par exemple que la somme des productions des unités égale une certaine demande totale en énergie. On a maintenant que la stratégie de gestion de chaque unité dépend de l'état de toutes les unités du parc. L'ajout d'une simple contrainte scalaire a donc augmenté de façon importante la complexité du problème d'optimisation.

Nous sommes pourtant face à un grand système de structure particulière (que nous qualifierons de structure en marguerite). Plusieurs méthodes ont proposé une manière de profiter de cette structure en marguerite pour baser les décisions sur une plus petite quantité d'information. Ainsi, la méthode de décomposition par agrégation de Turgeon (1980) cherche la stratégie de chaque unité comme une fonction de l'état local et d'une agrégation des états des autres unités en une quantité scalaire. Pour un grand système de ce type, Delebecque et Quadrat (1978) proposent une manière de rechercher la meilleure commande parmi les stratégies décentralisées, c'est-à-dire celles qui ne sont fonction que de "l'état local" du sous-système.

D'un autre côté, de nombreuses méthodes de décomposition dans le cadre stochastique sont basées sur une discrétisation de l'aléa sous la forme d'un arbre de scénarios. Ainsi, comme il a été présenté au §1.2.2 puis au §2.2.1, on approche la structure probabiliste du problème par une construction arborescente puis, une fois l'arbre construit, on transcrit les contraintes et l'objectif du problème sur cette structure. On peut alors faire appel aux méthodes de l'optimisation déterministe pour résoudre le problème discrétisé. En particulier, on peut faire appel aux techniques de décomposition classiques en optimisation déterministe. Cette approche a été étudiée par Carpentier *et al.* (1995), et fait l'objet du recueil de Higle et Sen (1996) ainsi que de celui de Ruszczyński et Shapiro (2003, Chapitre 3).

Cependant, comme il a été discuté au chapitre 2, pour des raisons indépendantes de la méthode de résolution mise en œuvre sur l'arbre, les techniques arborescentes ont le défaut de nécessiter, à précision donnée, un nombre de scénarios exponentiel en l'horizon de temps du problème. Par ailleurs, il faut bien réaliser que la difficulté liée à la taille de l'espace d'observation n'a pas encore été abordée à l'issue de la résolution du problème sur l'arbre de scénarios. En effet, on cherche à cette étape des commandes en boucle ouvert attachées à chaque nœud de l'arbre. ce ne serait que dans une phase ultérieure de synthèse d'une loi de commande permettant la construction d'une stratégie applicable à toute observation que la difficulté de la dimension de cette observation ressurgirait.

Nous présentons ici, pour un type de problème particulier, une approche basée sur la décomposition par les prix [1]. Ce type de problème est caractérisé par le fait que le système à commander est composé de sous-systèmes de plus petite taille ayant des dynamiques indépendantes et contribuant tous à la satisfaction d'une même contrainte. Ce modèle est

1. Encore une fois, d'autres types de décomposition existent (par les quantités, par prédiction) mais nous nous concentrerons ici sur la décomposition par les prix qui est la plus naturelle pour les applications qui nous intéressent. Nous nous en servirons ensuite dans le chapitre 4, où nous évoquerons aussi la décomposition par prédiction.

courant pour nombre d'applications, comme nous le montrons dans le chapitre 4.

L'originalité du travail présenté réside dans le fait que nous ne discrétisons pas a priori l'aléa mais laissons le choix aux sous-systèmes de résoudre leur problème local par la méthode de leur choix. On obtient alors des stratégies "locales-globales" reposant, pour chaque sous-système, sur la connaissance de l'état local de celui-ci et sur une variable d'information qui tente de résumer le reste du système. En effet, le multiplicateur de Lagrange (ou prix) qui nous permet de coordonner les sous-systèmes entre eux est, dans le contexte de la boucle fermée, lui aussi une variable aléatoire. Ainsi, il n'est généralement pas évident de résoudre directement les sous-problèmes, étant donné que nous ne connaissons rien a priori sur la mesurabilité de cette variable aléatoire. Ceci avait déjà été remarqué par Strugarek (2006) dans sa thèse. Nous proposons ici un cadre et une manière d'y parvenir de manière approchée, qui englobe le cas traité par Strugarek (2006). À notre connaissance, Barty et Roy (2007) furent les premiers à proposer le type d'approximation que nous énonçons ici, et dont une version préliminaire est décrite par Barty, Carpentier, et Girardeau (2010).

3.1 État de l'art

Nous allons spécifier le problème général (1.3) au cas qui nous intéresse ici. On considère un système dynamique ayant une structure particulière, que l'on qualifiera de décomposable. De manière informelle, on suppose que :

- le système est composé de sous-systèmes de petite taille ;
- les dynamiques des sous-systèmes sont indépendantes les unes des autres [2] ;
- la fonction objectif peut s'écrire comme une somme de coûts portant sur chaque sous-système indépendamment.

À titre d'exemple, on peut imaginer une collection de moyens de production ayant chacun des contraintes de fonctionnement propres (indépendantes unité par unité) et un coût de fonctionnement propre. On peut alors chercher à minimiser le coût de production total du système, qui est la somme des coûts de chaque unité. Nous nous plaçons à nouveau dans le cas de la mémoire parfaite, c'est-à-dire que la décision sur chaque unité est prise au regard de l'observation de tous les bruits passés ayant affecté le système global. Sans contrainte supplémentaire, le système se décompose naturellement unité par unité, ainsi que le problème de commande optimale associé. Autrement dit, on peut piloter de façon optimale le système en traitant chaque sous-système de manière parfaitement indépendante des autres.

Supposons maintenant qu'une contrainte supplémentaire couple le système à chaque pas de temps. Par exemple, on demande que la production totale du système dépasse une valeur donnée. Il est clair que la stratégie optimale de gestion du système n'aura pas, en général, la propriété d'être indépendante unité par unité. En effet, au niveau global, le décideur aura naturellement tendance à utiliser en premier lieu les moyens de production les moins onéreux, et à lancer progressivement les moyens plus coûteux jusqu'à satisfaire le niveau de production demandé. Ainsi, mettre en œuvre des stratégies locales, où la gestion de chaque unité ne se ferait que sur la base de l'information locale à l'unité, serait clairement sous-optimal du point de vue du critère économique global.

2. Les sous-systèmes peuvent être affectés par le même bruit exogène mais ne s'influencent pas les uns les autres.

Cependant, il est tentant de chercher à remplacer la résolution (lourde) du problème d'optimisation global par la résolution (éventuellement itérative) de problèmes locaux. Nous proposons dans ce chapitre une manière d'y parvenir, éventuellement au prix d'approximations sur la solution optimale que nous quantifierons.

Nous énonçons le problème qui nous intéresse au §3.1.1 et présentons la méthode de décomposition par les prix dans un cadre assez général. Au §3.1.2 nous présentons l'adaptation de la décomposition par les prix au cas de la boucle ouverte. Enfin, au §3.1.3, nous présentons l'approche par programmation stochastique et décomposition de la résolution de problèmes en boucle fermée.

3.1.1 Formulation du problème

Soit $n > 1$ un entier, les autres notations ayant été définies au chapitre 1, on considère le problème de commande optimale stochastique suivant :

$$\min_{\boldsymbol{X},\boldsymbol{U}} \quad \mathbb{E}\left(\sum_{t=0}^{T}\sum_{j=1}^{n} C_t^i\left(\boldsymbol{X}_t^j, \boldsymbol{U}_t^j\right)\right), \tag{3.1a}$$

$$\text{s.c.} \quad \boldsymbol{X}_{t+1}^j = f_t^j\left(\boldsymbol{X}_t^j, \boldsymbol{U}_t^j, \boldsymbol{W}_t\right), \qquad \forall t = 0,\dots,T-1, \forall j = 1,\dots,n, \tag{3.1b}$$

$$\boldsymbol{X}_0 = x_0, \tag{3.1c}$$

$$\sum_{j=1}^{n} g_t^j\left(\boldsymbol{U}_t^j\right) = 0, \qquad \forall t = 0,\dots,T, \tag{3.1d}$$

$$\boldsymbol{U}_t \preceq \mathcal{A}_t := \sigma\left\{\boldsymbol{W}_0,\dots,\boldsymbol{W}_t\right\}, \qquad \forall t = 0,\dots,T. \tag{3.1e}$$

Comme annoncé, nous avons affaire à n unités n'ayant pas d'influence les unes sur les autres. En l'absence de la contrainte (3.1d) le problème se décompose naturellement en n sous-problèmes, chacun portant sur l'une des unités [3]. C'est la raison pour laquelle cette contrainte, qui lie les unités entre elles, est dite *couplante*. Les méthodes que nous envisageons par la suite s'attachent à éliminer ce couplage en prenant garde, tant que possible, à ne pas modifier la solution du problème.

Remarque 3.1. Afin d'éviter certaines lourdeurs dans l'écriture, nous avons choisi de considérer une contrainte couplante (3.1d) ne faisant pas intervenir le bruit $\boldsymbol{W}_t$. Il n'y aurait cependant pas de difficulté théorique à considérer une contrainte où le bruit à l'instant t intervient, telle que : $\sum_{j=1}^{n} g_t^j(\boldsymbol{U}_t^j, \boldsymbol{W}_t) = 0$. En effet, l'ajout de $\boldsymbol{W}_t$ ne change pas la mesurabilité de la contrainte. Ce sera d'ailleurs le cas dans les applications numériques que nous considérerons.

Remarque 3.2. Contrairement aux modèles étudiés au chapitre 1, la commande à chaque instant est en "hasard-décision". En effet, comme énoncé dans la remarque 3.1, nous nous intéresserons souvent à des modèles dont la contrainte couplante fait intervenir le bruit à l'instant courant. Afin de pouvoir satisfaire une telle contrainte, il est évidemment nécessaire de connaître la valeur de ce bruit au moment de prendre la décision.

3. Ce ne serait pas le cas si le système nétait pas en mémoire parfaite, c'est-à-dire si, par exemple, la contrainte de mesurabilité (3.1e) imposait que, pour chaque unité i, la décision $\boldsymbol{U}_t^i$ soit mesurable par rapport à une partie du passé du bruit, disons à un abus de notation près $(\boldsymbol{W}_0^i,\dots,\boldsymbol{W}_t^i)$. À moins, bien sûr, de supposer l'indépendance entre les $\boldsymbol{W}^i$, pour $i = 1,\dots,n$.

3.1.2 Le cas de la boucle ouverte

Comme nous l'avons vu au §1.1.3, on parle de problème en boucle ouverte lorsque les décisions doivent être prises avant toute observation du système. Autrement dit, la tribu $\mathcal{A}_t$ représentant l'information disponible à l'instant t se réduit alors à la tribu grossière $\{\emptyset, \Omega\}$. Les variables de décision, qui sont a priori des variables aléatoires, éléments de $L^2(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{U})$, peuvent être représentées par des éléments de $\mathbb{U}$, du fait de la contrainte (3.1e). Le problème (3.1) se ramène alors à

$$\min_{\boldsymbol{X},u} \quad \mathbb{E}\left(\sum_{t=0}^{T}\sum_{j=1}^{n} C_t^j\left(\boldsymbol{X}_t^j, u_t^j\right)\right), \tag{3.2a}$$

$$\text{s.c.} \quad \boldsymbol{X}_{t+1}^j = f_t^j\left(\boldsymbol{X}_t^j, u_t^j, \boldsymbol{W}_t\right), \qquad \forall t = 0, \dots, T-1, \forall j = 1, \dots, n, \tag{3.2b}$$

$$\boldsymbol{X}_0 = x_0, \tag{3.2c}$$

$$\sum_{j=1}^{n} g_t^i\left(u_t^j\right) = 0, \qquad \forall t = 0, \dots, T. \tag{3.2d}$$

Il convient de remarquer à nouveau que les variables d'état du problème sont des variables intermédiaires : on peut facilement se ramener à un problème ne portant que sur la variable de décision u, en intégrant la dynamique (3.2b) sur tout l'horizon de temps. Ainsi, dans ce nouveau problème d'optimisation, l'aléa n'intervient plus que dans la fonction coût ; la contrainte couplante (3.2d) est quant à elle déterministe [4] et il en est de même des éventuels multiplicateurs de Lagrange associés. Ainsi, supposons qu'il existe un point-selle au Lagrangien (voir l'annexe A pour plus de détails) :

$$\mathcal{L}(u, \lambda) = \mathbb{E}\left(\sum_{t=0}^{T}\sum_{j=1}^{n}\left(C_t^j\left(\boldsymbol{X}_t^j, u_t^j\right) + \lambda_t \cdot g_t^j\left(u_t^j\right)\right)\right),$$

où la variable $\boldsymbol{X}$ est transportée par la dynamique (3.2b). Alors on peut se ramener à la résolution itérative de problèmes indexés par la valeur de λ de la forme :

$$\begin{aligned}
\min_{\boldsymbol{X},u} \quad & \mathbb{E}\left(\sum_{t=0}^{T}\sum_{j=1}^{n}\left(C_t^j\left(\boldsymbol{X}_t^j, u_t^j\right) + \lambda_t \cdot g_t^j\left(u_t^j\right)\right)\right), \\
\text{s.c.} \quad & \boldsymbol{X}_{t+1}^j = f_t^j\left(\boldsymbol{X}_t^j, u_t^j, \boldsymbol{W}_t\right), \qquad \forall t = 0, \dots, T-1, \forall j = 1, \dots, n, \\
& \boldsymbol{X}_0 = x_0,
\end{aligned}$$

qui se trouvent être décomposables en n sous-problèmes de plus petite taille. Ces derniers peuvent alors être résolus par des techniques classiques. Dans le cas différentiable, on peut par exemple utiliser un algorithme de type "gradient stochastique". Cela a fait l'objet de la thèse de Culioli (1987) (voir aussi l'article de Cohen et Culioli, 1990). On peut également utiliser les techniques d'échantillonnage (*Sample Average Approximation*) dont on trouvera les détails dans le recueil de Shapiro *et al.* (2009, chap. 4). Enfin, si on est capable de calculer de manière exacte l'espérance, on se ramène à un problème classique de programmation mathématique.

4. Remarquons que l'on ne pourrait pas ici placer le bruit $\boldsymbol{W}_t$ dans la contrainte couplante. La contrainte serait généralement irréalisable.

3.1.3 Arbres de scénarios pour traiter le cas de la boucle fermée

Un grand nombre de travaux permettant de discrétiser un problème d'optimisation stochastique en boucle fermée proviennent de la communauté *Stochastic Programming*. Nous ne rappelons pas ici cette méthodologie que nous avons déjà énoncée au §1.2.2 et étudié ensuite au §2.2. Rappelons-nous simplement qu'il s'agit de discrétiser la structure aléatoire du problème de départ à l'aide d'un arbre de scénarios afin de se ramener à un problème en dimension finie. Une fois le problème d'origine transcrit sur la structure arborescente, on tentera de le résoudre à l'aide d'une méthode ad hoc de l'optimisation déterministe. Lorsque le problème est de grande taille, on peut en particulier utiliser les techniques de décomposition connues en optimisation déterministe afin de résoudre le problème. Ce type d'approche est couramment utilisé et développé dans plusieurs ouvrages (Higle et Sen, 1996, Ruszczyński et Shapiro, 2003, chap. 3).

Cette classe de méthodes est très populaire, notamment du fait de sa simplicité de mise en œuvre, mais elle connaît deux inconvénients majeurs pour notre étude. D'une part, comme nous le montrons dans le chapitre 2, elle ne permet pas de garantir une précision suffisante pour des problèmes à plusieurs pas de temps avec un nombre raisonnable de scénarios. D'autre part, elle ne permet pas d'obtenir des stratégies mais seulement des commandes pour le premier pas de temps de l'intervalle d'étude. En effet, on n'introduit pas de notion d'état : les commandes obtenues en chacun des nœuds d'un arbre de scénario sont une fonction de tout le passé du bruit, matérialisé par le chemin allant du nœud courant à la racine de l'arbre. Cela rend pratiquement impossible l'utilisation de ces commandes, hormis de la première, c'est-à-dire celle à la racine de l'arbre, lors d'une phase de simulation. La seule manière d'utiliser cette technique dans un contexte à plusieurs pas de temps est de "reboucler", c'est-à-dire de reconstruire un nouvel arbre à chaque pas de temps lors de la simulation.

C'est pourquoi nous travaillons par la suite en adoptant un point de vue différent, reposant sur la notion de variable d'état ; c'est celui de la programmation dynamique ou des méthodes particulaires.

3.1.4 Un algorithme de décomposition général dans le cadre markovien

Nous cherchons maintenant à nous ramener au point de vue "programmation dynamique", qui est un cadre naturel en commande optimale stochastique. En effet, la stratégie optimale est a priori une fonction de tout le passé du bruit. Dès lors que le nombre de pas de temps devient important, cela constitue une information de bien trop grande taille sur laquelle baser des stratégies. En des termes plus mathématiques, l'espace de départ des fonctions que nous cherchons à optimiser grossit avec l'horizon de temps, ce qui rend rien que leur évaluation impossible en pratique (pour plus de détails sur la complexité des problèmes d'optimisation stochastique à plusieurs pas de temps, on pourra consulter l'article de Shapiro, 2006).

Le principe de programmation dynamique est un premier remède car il nous indique que, sous des hypothèses d'indépendance des bruits pas de temps par pas de temps, la variable $\boldsymbol{X}$ est une information suffisante sur laquelle baser la stratégie optimale. Or celle-ci ne grossit généralement pas avec le temps. Cependant, dans le cas de systèmes de grande taille[5], cela constitue encore une information de taille trop importante. Ainsi, la com-

5. Il est difficile de donner une dimension d'espace d'état précise à partir de laquelle le problème est

plexité de résolution de l'équation de programmation dynamique croît exponentiellement avec la dimension de l'espace d'état : c'est ce qui est couramment appelé la *malédiction de la dimension*.

Cela est une difficulté supplémentaire à celles rencontrées dans le cas de l'optimisation de grands systèmes en déterministe. Notons $\mathbb{X}^i$ (respectivement $\mathbb{U}^i$) l'espace d'arrivée de la variable $\boldsymbol{X}_t^i$ (respectivement $\boldsymbol{U}_t^i$), pour tout instant $t = 0, \dots, T$ et tout indice $i = 1, \dots, N$. Le principe de programmation dynamique nous indique en effet qu'il suffit de rechercher des stratégies de gestion de la forme :

$$\Phi_t^i \ : \mathbb{X}^1 \times \cdots \times \mathbb{X}^n \longrightarrow \mathbb{U}^i.$$

Cela signifie que la décision à appliquer à l'unité i à l'instant t est généralement une fonction de l'état de toutes les unités (d'où la *malédiction de la dimension*). Il est généralement inespéré de pouvoir se ramener à des stratégies "décentralisées" de la forme :

$$\Phi_t^i \ : \mathbb{X}^i \longrightarrow \mathbb{U}^i.$$

Revenant au problème (3.1), la théorie énoncée par Cohen (1984) étant faite dans des espaces de Hilbert, il n'y a pas de difficulté théorique empêchant d'énoncer un algorithme de décomposition par les prix du même type que dans le cas déterministe, puisque nous avons pris la précaution de nous placer avec des variables aléatoires de carré intégrable. La contrainte couplante (3.1d) étant stochastique, on introduit donc, sous les hypothèses nécessaires de qualification des contraintes, le multiplicateur de Lagrange $\boldsymbol{\lambda}_t$. Il faut noter que celui-ci est un processus stochastique qui a même mesurabilité que la contrainte à laquelle il est associé : il est donc $\mathcal{A}_t$-mesurable. Suivons le même cheminement que dans le cas de la boucle ouverte et supposons qu'il existe un point-selle au Lagrangien :

$$\mathcal{L}\left(\boldsymbol{U}, \boldsymbol{\lambda}\right) = \mathbb{E}\left(\sum_{t=0}^{T}\sum_{j=1}^{n}\left(C_t^j\left(\boldsymbol{X}_t^j, \boldsymbol{U}_t^j\right) + \boldsymbol{\lambda}_t \cdot g_t^j\left(\boldsymbol{U}_t^j\right)\right)\right),$$

où la variable $\boldsymbol{X}$ est transportée par la dynamique (3.1b). Alors on peut se ramener à la résolution itérative de problèmes de la forme :

$$\min_{\boldsymbol{X}^j, \boldsymbol{U}^j} \quad \mathbb{E}\left(\sum_{t=0}^{T}\left(C_t^j\left(\boldsymbol{X}_t^j, \boldsymbol{U}_t^j\right) + \boldsymbol{\lambda}_t \cdot g_t^j\left(\boldsymbol{U}_t^j\right)\right)\right), \tag{3.3a}$$

$$\text{s.c.} \quad \boldsymbol{X}_{t+1}^j = f_t^j\left(\boldsymbol{X}_t^j, \boldsymbol{U}_t^j, \boldsymbol{W}_t\right), \qquad \forall t = 0, \dots, T-1, \tag{3.3b}$$

$$\boldsymbol{X}_0 = \boldsymbol{W}_0, \tag{3.3c}$$

$$\boldsymbol{U}_t^j \preceq \mathcal{A}_t, \qquad \forall t = 0, \dots, T. \tag{3.3d}$$

et énoncer l'algorithme 1 de décomposition par les prix. Sous les hypothèses classiques de convexité et de gradient lipschitzien sur la fonction objectif du problème (3.3), ainsi que de linéarité de la dynamique, l'algorithme converge pour une valeur du pas ρ suffisamment petite (voir Ekeland et Temam, 1999, pour plus de détails). L'interprétation de l'algorithme est la même que dans le cas déterministe : à chaque itération, le coordonnateur envoie aux unités un prix $\boldsymbol{\lambda}^k$, auquel la production de chaque unité sera rémunérée. Chaque unité calcule alors la quantité de production qui minimise la différence entre son

considéré de grande taille. Cela dépend beaucoup du problème en question. On peut considérer qu'à partir de la dimension 5, la résolution de l'équation de programmation dynamique devient trop ardue.

Entrées : Un prix $\boldsymbol{\lambda}^1$ arbitraire.
Sorties : Le prix $\boldsymbol{\lambda}^K$ et des stratégies $\boldsymbol{U}^{j,K}$.
pour $k = 1, \dots, K$ **faire**
  pour $j = 1, \dots, n$ **faire**
    Résoudre le sous-problème (3.3) et ainsi obtenir une commande $\boldsymbol{U}^{j,k}$.
  fin
  Mettre à jour les prix :

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho \times \sum_{j=1}^{n} g^j \left(\boldsymbol{U}^{j,k} \right).$$

fin

Algorithme 1 : Algorithme de décomposition par les prix en boucle fermée.

coût de production et la rémunération offerte par le coordonnateur, et renvoie cette production à ce dernier, qui ajuste ensuite le prix au regard de la satisfaction de la contrainte couplante.

Cependant deux questions se posent :

1. Comment effectuer la mise à jour des prix, qui concerne des variables aléatoires ? En général, nous ne connaissons pas de manière analytique les lois des variables aléatoires impliquées dans le problème. En revanche, nous sommes capables d'effectuer des tirages de ces variables aléatoires. Ainsi, pour ce qui est de la mise à jour des prix, il est naturel de proposer une opération scénario par scénario. On effectue avant le démarrage du processus itératif un certain nombre de tirages de scénarios des bruits. Puis, à chaque itération, étant donnée la loi de commande courante, on effectue la mise à jour du prix pour chaque scénario à l'aide de la commande évaluée sur ce même scénario.
2. Comment résoudre les sous-problèmes ? La contrainte de mesurabilité (3.3d) qui apparaît dans chacun des sous-problèmes est la même que dans le problème de départ. La stratégie de gestion associée à l'unité j dépend ainsi a priori de tout le passé des bruits. Nous aimerions nous ramener, à l'aide d'un argument d'indépendance en temps des bruits, à une stratégie en *feedback* sur un état, de plus petite taille que le passé des bruits. Mais il est en général faux d'affirmer que les variables aléatoires $\boldsymbol{\lambda}_0, \dots, \boldsymbol{\lambda}_T$ sont indépendantes entre elles. Plus généralement, on ne sait rien sur la dynamique de ces prix à telle ou telle itération de l'algorithme, aussi bien qu'à l'optimum. Il n'est donc en général pas possible de résoudre les sous-problèmes par programmation dynamique sur un état de petite taille.

3.1.5 Cas particulier de la résolution des sous-problèmes

Strugarek (2006) introduit une instance du problème (3.1) pour lequel il est capable d'exhiber la dynamique du prix optimal. L'ordre de cette dynamique étant de petite taille, il est dès lors capable de résoudre les sous-problèmes de manière efficace par programmation dynamique. Nous rappelons maintenant son résultat.

Le problème est inspiré de la gestion de la production d'un parc électrique. Prenons un producteur d'énergie disposant de n centrales de production soumises à des apports aléatoires $\boldsymbol{A}_{t+1}^j$ au pas de temps t, pour l'unité j. On écrira $\boldsymbol{A}_{t+1}$ pour désigner la collection

de tous les apports au pas de temps t. Chaque unité a de plus un coût de production quadratique en sa production. Le producteur est contraint de satisfaire une demande en énergie aléatoire $\boldsymbol{D}_t$ à chaque pas de temps t. On suppose enfin être en information parfaite : la décision de production pour chacune des centrales à l'instant t est basée sur la connaissance des réalisations de tous les bruits (demande, apports) jusqu'au pas de temps t. En particulier, au moment où la décision est prise, le décideur connaît la réalisation de la demande pour le pas de temps courant. En revanche, il ne connaît pas la réalisation des apports qui affecteront les lacs au pas de temps courant. On dit que le problème est en "hasard-décision" sur la demande, et en "décision-hasard" sur les apports.

Soit, donc, le problème de commande optimale stochastique suivant :

$$\min_{\boldsymbol{X},\boldsymbol{U}} \quad \mathbb{E}\left(\sum_{t=1}^{T-1}\sum_{j=1}^{n} c_j \frac{\left(\boldsymbol{U}_t^j\right)^2}{2} + \frac{\gamma_j}{2}\left(\boldsymbol{X}_t^j - x_1^j\right)^2\right), \tag{3.4a}$$

$$\text{s.c.} \quad \boldsymbol{X}_{t+1}^j = \boldsymbol{X}_t^j + \boldsymbol{A}_{t+1}^j - \boldsymbol{U}_t^j, \qquad \forall t = 1,\dots,T-1, \forall j = 1,\dots,n, \tag{3.4b}$$

$$\sum_{j=1}^{n} \boldsymbol{U}_t^j = \boldsymbol{D}_t, \qquad \forall t = 1,\dots,T-1, \tag{3.4c}$$

$$\boldsymbol{U}_t \preceq \sigma\left\{\boldsymbol{D}_s, s \le t\,;\, \boldsymbol{A}_s, s \le t\right\}. \tag{3.4d}$$

On peut alors montrer le résultat suivant :

Proposition 3.1 (Strugarek, 2006)**.** *Si les variables aléatoires $(\boldsymbol{D}_t, \boldsymbol{A}_t)_{t=1,\dots,T}$ sont indépendantes pas de temps par pas de temps, et si il existe $\alpha > 0$ tel que $\gamma_j = \alpha c_j$, pour tout $j = 1,\dots,n$, alors le multiplicateur optimal $\boldsymbol{\lambda}$ associé à l'ensemble des contraintes couplantes* (3.4c) *est donné par l'équation dynamique :*

$$\boldsymbol{\lambda}_1 = \frac{1}{\sum_{j=1}^n \frac{1}{c_j}}\left(\boldsymbol{D}_1(1-\alpha) - \alpha\sum_{s=2}^{T}\mathbb{E}(\boldsymbol{A}_s) - \alpha\sum_{s=2}^{T-1}\mathbb{E}(\boldsymbol{D}_s)\right),$$

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \frac{1}{\sum_{j=1}^n \frac{1}{c_i}}\Big[\boldsymbol{D}_{t+1}(1+\alpha) - \boldsymbol{D}_t - \alpha\mathbb{E}(\boldsymbol{D}_{t+1}) - \alpha\left(\boldsymbol{A}_{t+1} - \mathbb{E}(\boldsymbol{A}_{t+1})\right)\Big], \qquad \forall t = 1,\dots,T-2.$$

Ainsi, en incorporant la mémoire du prix ainsi que de la demande sur un pas de temps dans l'état, on peut résoudre les sous-problèmes issus de la dualisation de la contrainte couplante dans le problème (3.4) par programmation dynamique en dimension 3. On a de cette manière remplacé la résolution d'un problème de dimension n par la résolution de n problèmes de dimension 3. L'équation de la programmation dynamique pour le sous-problème i s'écrit :

$$V_T^j(x,\lambda,d) = \frac{\gamma_j}{2}\left(x - x_1^j\right)^2,$$

$$V_t^j(x,\lambda,d) = \begin{cases} \min\limits_{u} \quad \mathbb{E}\left(c_j\dfrac{u^2}{2} + \lambda\left(\dfrac{1}{n}d - u\right) + V_{t+1}^j\left(\boldsymbol{X}_{t+1}^j, \boldsymbol{\lambda}_{t+1}, \boldsymbol{D}_{t+1}\right)\right), \\ \text{s.c.} \quad \boldsymbol{X}_{t+1}^j = x + \boldsymbol{A}_{t+1}^j - u, \\ \qquad \boldsymbol{\lambda}_{t+1} = \lambda + \frac{1}{\sum_{j=1}^n \frac{1}{c_j}}\Big[\boldsymbol{D}_{t+1}(1+\alpha) - d - \alpha\mathbb{E}(\boldsymbol{D}_{t+1}) \\ \qquad\qquad -\alpha\left(\boldsymbol{A}_{t+1} - \mathbb{E}(\boldsymbol{A}_{t+1})\right)\Big]. \end{cases}$$

Ainsi les stratégies locales sont elles fonction de l'état local, du prix, et de la demande, tous trois étant des scalaires. L'enseignement que nous pouvons tirer de cet exemple est le suivant. Bien que, dans le cas de la boucle fermée, le multiplicateur associé à la contrainte couplante dépende est général de toute le passé du bruit, il existe des cas (nous avons même ici une preuve constructive) où la mesurabilité de ces multiplicateurs par rapport au passé du bruit a une mémoire finie. Pour résoudre de manière efficace les sous-problèmes dans le cadre d'une décomposition par les prix, il s'agit de trouver la statistique minimale permettant de caractériser ce prix. Dans le cas décrit ci-dessus, la mémoire du prix et de la demande sur un pas de temps constituent cette statistique minimale. Lorsqu'une relation telle que celle de la proposition 3.1 ne peut pas être obtenue de manière analytique, on peut imaginer avoir recours à des outils statistiques afin d'exhiber ce type de relation par l'expérience. C'est ce que nous proposons au §3.2.

Il convient de noter que le résultat de la proposition 3.1 a peu de chances de se généraliser. L'hypothèse demandant une proportionnalité entre les coûts, notamment, ne semble pas avoir de sens pratique. En toute généralité, il est fort probable que le prix à un instant donné dépende effectivement de tout le passé du bruit.

3.2 Résolution des sous-problèmes en commande optimale stochastique

Nous avons vu que la résolution des sous-problèmes issus de la décomposition d'un problème de commande optimale stochastique, c'est-à-dire l'obtention de stratégies de gestion pour chaque unité, est pratiquement aussi difficile que pour le problème de départ. En effet, la stratégie de gestion d'une unité dépend en général de toute l'information disponible. Dans le cadre markovien, elle dépend des états de toutes les unités. Nous nous retrouvons alors confrontés à la *malédiction de la dimension* que connaît la programmation dynamique.

Dans un premier travail (Barty *et al.*, 2010), nous proposions de fixer a priori une forme pour la dynamique du multiplicateur (par exemple un processus autorégressif d'ordre 1), ce qui permettait, si la dynamique n'était pas de trop grande taille, de résoudre les sous-problèmes par programmation dynamique. Nous obtenions alors des résultats numériques prometteurs.

Nous présentons ici une approche que nous pensons être plus générale. Elle permet de plus d'interpréter l'approximation que nous effectuons en terme de satisfaction des contraintes que nous dualisons. Sa mise en œuvre pratique est également facilitée. Nous appelons cet algorithme *Dual Approximate Dynamic Programming* (DADP) en référence à la programmation dynamique approximée qui propose de choisir une forme a priori pour la fonction de Bellman ; nous choisissons ici une forme (ou plutôt une mesurabilité) a priori pour la variable duale qui nous permet de décomposer le problème.

3.2.1 Résolution approchée des sous-problèmes par programmation dynamique

Nous remarquons que la seule variable "gênante" au sein des sous-problèmes est le multiplicateur, du fait qu'il dépend en général de tout le passé du bruit. Nous avons vu, au travers d'un exemple, qu'il est parfois possible de représenter correctement ce prix à partir de peu d'information supplémentaire, et donc de garder un espace d'état de

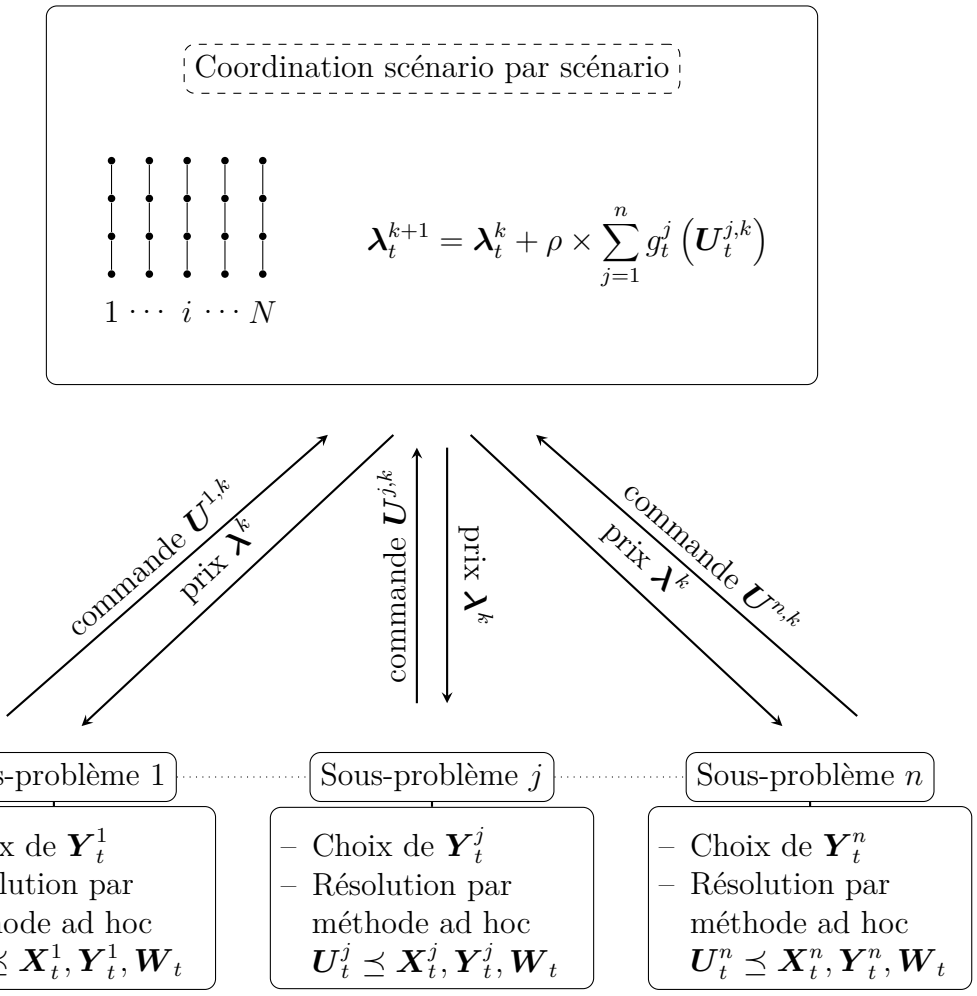

FIGURE 3.1 – Schéma d'un algorithme général de décomposition par les prix en boucle fermée.

taille raisonnable dans les sous-problèmes. La méthodologie que nous proposons consiste à remplacer le prix issu du coordonnateur par son espérance conditionnelle par rapport à ce que nous appellerons une variable d'information. Celle-ci est choisie au moment de la résolution ; ce n'est pas une variable d'optimisation. Plus cette variable, fonction du passé des bruits, "contiendra" d'information, plus le prix sera bien représenté, mais plus la résolution du sous-problème par programmation dynamique sera difficile. Nous décrivons la démarche globale dans la figure 3.1.

On se donne donc une variable $\boldsymbol{Y}_t^j$ mesurable par rapport à $\mathcal{A}_t$, à chaque instant t et pour chaque sous-problème j, qui ne doit pas être influencée par les variables d'optimisation. On propose de remplacer la résolution du problème (3.3) par la résolution du

problème suivant.

$$\min_{\boldsymbol{X}^j, \boldsymbol{U}^j} \quad \mathbb{E}\left(\sum_{t=0}^{T}\left(C_t^j\left(\boldsymbol{X}_t^j, \boldsymbol{U}_t^j\right) + \mathbb{E}\left(\boldsymbol{\lambda}_t \;\middle|\; \boldsymbol{Y}_t^j\right) \cdot g_t^j\left(\boldsymbol{U}_t^j\right)\right)\right), \tag{3.5a}$$

$$\text{s.c.} \quad \boldsymbol{X}_{t+1}^j = f_t^j\left(\boldsymbol{X}_t^j, \boldsymbol{U}_t^j, \boldsymbol{W}_t\right), \qquad \forall t = 0, \dots, T-1, \tag{3.5b}$$

$$\boldsymbol{X}_0 = \boldsymbol{W}_0, \tag{3.5c}$$

$$\boldsymbol{U}_t^j \preceq \mathcal{A}_t, \qquad \forall t = 0, \dots, T. \tag{3.5d}$$

On envisage alors deux cas. Dans le premier, on choisit la variable $\boldsymbol{Y}_t^j$ comme fonction uniquement du bruit $\boldsymbol{W}_t$ à l'instant t. On peut alors écrire un principe de programmation dynamique pour le problème (3.5) :

$$V_t^j(x) = \mathbb{E}\left(\min_u C_t^j(x,u) + \mathbb{E}\left(\boldsymbol{\lambda}_t \;\middle|\; \boldsymbol{Y}_t^j\right) \cdot g_t^j(u) + V_{t+1}^j\left(f_t^j(x,u,\boldsymbol{W}_t)\right)\right).$$

Lors de la résolution de l'équation de programmation dynamique, on observe que la commande à l'instant t est en *feedback* sur le stock local $\boldsymbol{X}_t^j$, le bruit $\boldsymbol{W}_t$ et l'information $\boldsymbol{Y}_t^j$, qui elle-même est une fonction du bruit $\boldsymbol{W}_t$. En revanche, la fonction de Bellman ne dépend que du stock local. Cette différence provient simplement du fait que l'on est dans le cadre "hasard-décision".

Du fait de l'indépendance des variables d'information $\boldsymbol{Y}_t^j$ pas de temps par pas de temps, on a donc à résoudre des équations de programmation dynamique où la dimension est celle du sous-système (celle de $\mathbb{X}^j$). Donnons trois exemples de choix d'une telle variable d'information.

Exemple 3.1 (Maximum d'information). On peut choisir de mettre toute le bruit de l'instant t dans la variable d'information $\boldsymbol{Y}_t^j$. Rappelons que dans le bruit, une partie peut être "locale" (apport dans un stock d'eau) et une autre partie peut être "globale" (demande en énergie sur le réseau). Ainsi, inclure le bruit dans la variable d'information permet éventuellement de fournir une information de type "global" au sous-système. Notons qu'inclure tout le bruit à l'instant t dans la variable d'information n'est possible en pratique que lorsque celui-ci n'est pas de trop grande taille. On a en effet à calculer une espérance conditionnelle dont le conditionnement est $\boldsymbol{W}_t$. Ce calcul est soumis lui aussi à la malédiction de la dimension.

Exemple 3.2 (Minimum d'information). À l'inverse, on peut choisir $\boldsymbol{Y}_t^j = 0$ ou toute autre constante. On approche alors le prix à chaque instant par son espérance. Si le stock local est de taille raisonnable, l'équation de programmation dynamique qui en découle est alors soluble et on obtient une commande qui correspond à une vision du prix en moyenne.

Exemple 3.3 (Entre les deux). On peut enfin se donner $\boldsymbol{Y}_t^j$ sous la forme $h_t^j(\boldsymbol{W}_t)$. En pratique, ce choix sera généralement guidé par l'intuition que l'on a de l'information importante pour "expliquer" le multiplicateur optimal. Il s'agit de faire un compromis entre la quantité d'information suffisante pour expliquer raisonnablement le processus de prix et la complexité du calcul d'espérance conditionnelle dans (3.5a).

Le deuxième cas est plus général. On peut chercher à retenir de l'information au cours du temps. Autrement dit, on peut choisir une variable d'information ayant une dynamique markovienne, c'est-à-dire sous la forme $\boldsymbol{Y}_{t+1}^j = h_t^j(\boldsymbol{Y}_t^j, \boldsymbol{W}_{t+1})$. Dans ce cas, afin de pouvoir écrire l'équation de programmation dynamique, on doit augmenter l'état afin d'y inclure, à l'instant t, la mémoire nécessaire au calcul de la variable d'information $\boldsymbol{Y}_t$. Ainsi, la

fonction de Bellman associée au sous-problème j dépend maintenant à la fois de $\boldsymbol{X}_t^j$ et de $\boldsymbol{Y}_{t-1}^j$. L'équation de programmation dynamique s'écrit :

$$V_t^j(x,y) = \mathbb{E}\Big(\min_u \quad C_t^j(x,u) + \mathbb{E}\left(\boldsymbol{\lambda}_t \;\middle|\; \boldsymbol{Y}_t^j\right) \cdot g_t^j(u) + V_{t+1}^j\left(f_t^j(x,u,\boldsymbol{W}_t), \boldsymbol{Y}_t^j\right)\Big),$$
$$\text{s.c.} \quad \boldsymbol{Y}_t^j = h_{t-1}^j(y, \boldsymbol{W}_t).$$

Lors de la résolution de cette équation de programmation dynamique, on obtient une commande en *feedback* sur le stock local $\boldsymbol{X}_t^j$, le bruit courant $\boldsymbol{W}_t$ et l'information $\boldsymbol{Y}_{t-1}^j$ conservée du pas de temps précédent. Ce décalage d'indice entre information et stock provient du cadre "hasard-décision" : à l'instant t, l'information à partir de laquelle nous prenons la décision est une conjonction de l'information retenue (qui porte l'indice $t-1$) et du bruit observé à l'instant courant $\boldsymbol{W}_t$.

Exemple 3.4 (Maximum d'information). Le choix de $\boldsymbol{Y}_t^j = (\boldsymbol{W}_0, \ldots, \boldsymbol{W}_t)$ rentre dans ce cadre. On a alors $\mathbb{E}\left(\boldsymbol{\lambda}_t \;\middle|\; \boldsymbol{Y}_t^j\right) = \boldsymbol{\lambda}_t$. On a donc parfaitement approché le prix, mais l'équation de programmation dynamique n'est pas soluble en pratique pour les raisons déjà citées.

Exemple 3.5 (Strugarek, 2006). Dans le cas particulier rappelé au §3.1.5, on avait réussi à expliciter une variable d'information qui se trouvait être égale au multiplicateur optimal. Pour reprendre les notations du §3.1.5, en choisissant comme variable d'information $(\boldsymbol{Y}_t, \boldsymbol{D}_t)$ avec :

$$\boldsymbol{Y}_1 = \frac{1}{\sum_{j=1}^n \frac{1}{c_j}} \left(\boldsymbol{D}_1 (1-\alpha) - \alpha \sum_{s=2}^{T} \mathbb{E}(\boldsymbol{A}_s) - \alpha \sum_{s=2}^{T-1} \mathbb{E}(\boldsymbol{D}_s) \right),$$
$$\boldsymbol{Y}_{t+1} = \boldsymbol{Y}_t + \frac{1}{\sum_{j=1}^n \frac{1}{c_j}} \Big[\boldsymbol{D}_{t+1}(1+\alpha) - \boldsymbol{D}_t - \alpha \mathbb{E}(\boldsymbol{D}_{t+1})$$
$$- \alpha (\boldsymbol{A}_{t+1} - \mathbb{E}(\boldsymbol{A}_{t+1})) \Big], \qquad \forall t = 1, \ldots, T-2,$$

on se retrouve dans le cas particulier où $\mathbb{E}\left(\boldsymbol{\lambda}_t \;\middle|\; \boldsymbol{Y}_t^j\right) = \boldsymbol{\lambda}_t$, avec la variable d'information $\boldsymbol{Y}_t^j$ de petite dimension (il fallait retenir la dernière valeur de la demande et la valeur de $\boldsymbol{Y}_{t-1}^j$, qui était de dimension 1).

On a proposé une manière de projeter le multiplicateur envoyé par le coordonnateur de sorte à ce que le sous-problème soit soluble par programmation dynamique. Pour pouvoir appliquer de manière efficace cette méthodologie, deux problèmes se posent :

1. il faut identifier l'information qui permettra d'estimer le prix le plus précisément possible ;
2. il faut être capable d'effectuer le mieux possible les calculs d'espérance conditionnelle du type $\mathbb{E}\left(\boldsymbol{\lambda}_t \;\middle|\; \boldsymbol{Y}_t^j\right)$.

Il s'agit maintenant de comprendre les effets des approximations effectuées au niveau global.

3.2.2 Résultats théoriques du point de vue global

L'approximation que nous proposons doit permettre de calculer numériquement des stratégies pour chacun des sous-systèmes. Selon le choix de la variable d'information

supplémentaire à l'état local, on imagine bien que certaines stratégies seront meilleures que d'autres (du point de vue de la satisfaction de la contrainte couplante, ou encore du coût dual). On essaie ici de quantifier ces approximations.

Dorénavant, nous considérons que la variable d'information est la même pour tous les sous-sytèmes. Notons-la $\boldsymbol{Y}_t$ et définissons les espaces vectoriels $\mathcal{Y}_t := \{\boldsymbol{\lambda}_t \in L^2(\Omega, \mathcal{A}, \mathbb{P}) : \boldsymbol{\lambda}_t \preceq \boldsymbol{Y}_t\}$, pour tout instant $t = 1, \dots, T$. L'algorithme que nous proposons revient à répéter les opérations suivantes :

- étant donné un processus de prix, résoudre les sous-problèmes avec la projection de ce processus de prix sur $\mathcal{Y}_0 \times \cdots \times \mathcal{Y}_T$;
- mettre à jour le processus de prix par une formule de gradient.

Ainsi, on peut interpréter l'algorithme proposé comme un algorithme de gradient projeté appliqué à la maximisation de la fonction duale du problème, ce qui revient à considérer le problème dual approché :

$$\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{X},\boldsymbol{U}} \quad \mathbb{E}\left(\sum_{t=0}^{T}\left(\sum_{j=1}^{n} C_t^i\left(\boldsymbol{X}_t^j, \boldsymbol{U}_t^j\right) + \boldsymbol{\lambda}_t \cdot \sum_{j=1}^{n} g_t^j\left(\boldsymbol{U}_t^j\right)\right)\right), \tag{3.6a}$$

$$\text{s.c.} \quad \boldsymbol{X}_{t+1}^j = f_t^j\left(\boldsymbol{X}_t^j, \boldsymbol{U}_t^j, \boldsymbol{W}_t\right), \qquad \forall t = 0, \dots, T-1, \forall j = 1, \dots, n, \tag{3.6b}$$

$$\boldsymbol{X}_0 = x_0, \tag{3.6c}$$

$$\boldsymbol{U}_t \preceq \mathcal{A}_t, \qquad \forall t = 0, \dots, T, \tag{3.6d}$$

$$\boldsymbol{\lambda}_t \preceq \boldsymbol{Y}_t, \qquad \forall t = 0, \dots, T. \tag{3.6e}$$

Or, dans le produit scalaire $\langle a, b\rangle$, si a appartient à un sous-espace, alors la composante de b orthogonale à ce sous-espace donne zéro dans le produit scalaire et est donc inutile. Autrement dit, le multiplicateur a ne peut "observer" que la composante de b qui est dans le même sous-espace que a, mais pas la composante dans le sous-espace orthogonal. Ainsi, si le Lagrangien associé à ce problème a un point-selle (voir l'annexe A et l'ouvrage de Ekeland et Temam, 1999, pour plus de détails sur l'existence d'un point-selle au Lagrangien), alors le problème (3.6) a même valeur que le problème primal associé :

$$\min_{\boldsymbol{X},\boldsymbol{U}} \quad \mathbb{E}\left(\sum_{t=0}^{T}\sum_{j=1}^{n} C_t^j\left(\boldsymbol{X}_t^j, \boldsymbol{U}_t^j\right)\right), \tag{3.7a}$$

$$\text{s.c.} \quad \boldsymbol{X}_{t+1}^j = f_t^j\left(\boldsymbol{X}_t^j, \boldsymbol{U}_t^j, \boldsymbol{W}_t\right), \qquad \forall t = 0, \dots, T-1, \forall j = 1, \dots, n, \tag{3.7b}$$

$$\boldsymbol{X}_0 = \boldsymbol{W}_0, \tag{3.7c}$$

$$\mathbb{E}\left(\sum_{j=1}^{n} g_t^j\left(\boldsymbol{U}_t^j\right) \,\middle|\, \boldsymbol{Y}_t\right) = 0, \qquad \forall t = 0, \dots, T, \tag{3.7d}$$

$$\boldsymbol{U}_t \preceq \mathcal{A}_t, \qquad \forall t = 0, \dots, T. \tag{3.7e}$$

On a donc remplacé une contrainte presque-sûre par une contrainte en espérance conditionnellement à la variable d'information choisie. Il apparait ici clairement que si l'on choisit à travers la variable d'information $\boldsymbol{Y}_t$ de retenir tout le passé des bruits, on récupère la contrainte de départ et on résout le problème initial. C'est le cas de l'exemple 3.4 et ce n'est généralement pas possible en pratique. À l'inverse, ne mettre aucune information dans la variable $\boldsymbol{Y}_t$ revient à satisfaire la contrainte couplante en espérance seulement. C'est le cas de l'exemple 3.2.

Toute la difficulté est donc de trouver la variable d'information $\boldsymbol{Y}_t$ qui permettra de satisfaire au mieux la contrainte couplante tout en évitant de rendre la résolution des sous-problèmes trop difficile.

3.2.3 Convergence

Remarquons que nous n'avons parlé que de la valeur des problèmes duaux et primaux. La convergence des stratégies obtenues par l'algorithme DADP est quant à elle une application directe du théorème (A.3). Ainsi, sous les hypothèses classiques de convexité et de continuité qui sont précisées dans ce théorème, on a que l'algorithme de décomposition stochastique génère une suite de stratégies qui converge vers la stratégie optimale du problème (3.7). Le lecteur pourra se référer à Ekeland et Temam (1999) ou à Cohen (2004) pour une présentation plus générale dans le cadre du Principe du Problème Auxiliaire.

Ce premier résultat nécessite cependant de faire une hypothèse de forte convexité sur la fonction coût qui peut être restrictive pour certaines applications, où la fonction coût est seulement convexe. Dans ce cas, la fonction duale n'est plus différentiable et il serait donc maladroit de lui appliquer brutalement un algorithme de (sous)-gradient à pas fixe. De plus, même si l'on parvient à converger dans le dual, on trouve associé au multiplicateur vers lequel on converge un continuum de solutions primales [6] dont la la plupart ne satisfont pas la contrainte couplante. Dans ce cas, on a classiquement recours à la régularisée de Yosida-Moreau du Lagrangien, qui récupère la propriété de différentiabilité, et que l'on appelle Lagrangien Augmenté (voir Cohen, 2000, Chapitre 7). Le Lagrangien augmenté (ou *multiplier method* en anglais) a été introduit pour surmonter les sauts de dualité qui apparaissent dans le cas non-convexe (voir Bertsekas, 1982). Dans le cas convexe, mais non strictement convexe, il va régulariser le problème dual afin de "stabiliser" la convergence. Décrivons ce nouveau Lagrangien pour le problème (3.7) :

$$\mathcal{L}_c\left(\boldsymbol{U},\boldsymbol{\lambda}\right) = \mathbb{E}\left[\sum_{t=0}^{T}\left(\sum_{j=1}^{n}\left(C_t^j\left(\boldsymbol{X}_t^j,\boldsymbol{U}_t^j\right)+\boldsymbol{\lambda}_t\cdot g_t^j\left(\boldsymbol{U}_t^j\right)\right)+\frac{c}{2}\left(\mathbb{E}\Big(\sum_{j=1}^{n} g_t^j\left(\boldsymbol{U}_t^j\right)\Big|\boldsymbol{Y}_t\Big)\right)^2\right)\right].$$

L'inconvénient du Lagrangien augmenté est que le terme quadratique couple à nouveau toutes les unités entre elles. Cohen et Zhu (1984) proposent de linéariser ce terme au voisinage de l'itéré courant afin de revenir à une structure additive. On obtient alors, pour chaque unité j à l'itération $k+1$, le sous-problème :

$$\begin{aligned}
\min_{\boldsymbol{X}^j,\boldsymbol{U}^j}\quad & \mathbb{E}\Bigg[\sum_{t=0}^{T}\Bigg(C_t^j\left(\boldsymbol{X}_t^j,\boldsymbol{U}_t^j\right)+g_t^j\left(\boldsymbol{U}_t^j\right)\cdot\mathbb{E}\left(\boldsymbol{\lambda}_t^k\;\middle|\;\boldsymbol{Y}_t\right)\\
&\qquad\qquad +c\cdot\mathbb{E}\left(\sum_{j'=1}^{n} g_t^{j'}\left(\boldsymbol{U}_t^{j',k}\right)\;\middle|\;\boldsymbol{Y}_t\right)\cdot\mathbb{E}\left(g_t^j\left(\boldsymbol{U}_t^j\right)-g_t^j\left(\boldsymbol{U}_t^{j,k}\right)\;\middle|\;\boldsymbol{Y}_t\right)\Bigg)\Bigg],\\
\text{s.c.}\quad & \boldsymbol{X}_{t+1}^j = f_t^j\left(\boldsymbol{X}_t^j,\boldsymbol{U}_t^j,\boldsymbol{W}_t\right), \qquad \forall t=0,\dots,T-1,\\
& \boldsymbol{X}_0=\boldsymbol{W}_0,\\
& \boldsymbol{U}_t^j \preceq \mathcal{A}_t, \qquad \forall t=0,\dots,T.
\end{aligned}$$

On remarque qu'afin d'utiliser un Lagrangien augmenté linéarisé, il est nécessaire d'effectuer un calcul d'espérance conditionnelle portant sur la quantité $\sum_{j'=1}^{n} g_t^{j'}(\boldsymbol{U}_t^{j',k})$. Comme pour le processus de prix $\boldsymbol{\lambda}$, celui-ci peut se faire à part, avant la résolution à proprement parler du sous-problème. Elle n'est en soit pas plus complexe que celle qui concerne le multiplicateur $\boldsymbol{\lambda}$. La résolution du sous-problème peut se faire de la même manière que dans le cas du Lagrangien simple, par programmation dynamique par exemple, avec un état de même taille : l'ajout du terme quadratique ne nous oblige donc pas à augmenter l'état

6. C'est le sur-différentiel de la fonction duale en le multiplicateur vers lequel on a convergé.

dans les sous-problèmes. L'étape de mise à jour des multiplicateurs par le coordonnateur est quant à elle identique au cas du Lagrangien simple.

3.3 Conclusion

La résolution de problèmes de commande optimale stochastique de grande taille présente une difficulté importante : une stratégie de gestion optimale d'un grand système dépend par définition d'un grand nombre de paramètres qui constituent son état, et chercher une solution sous la forme de *feedbacks* décentralisés, c'est-à-dire de stratégies qui pour chaque unité ne dépendent que de "l'état local" à l'unité, est en général sous-optimal.

Nous avons proposé une méthode de décomposition adaptée au cadre stochastique, appelée *Dual Approximate Dynamic Programming* (DADP) qui, via une étape de projection du multiplicateur de Lagrange, ouvre la porte à une résolution efficace de tels problèmes. Toute l'efficacité d'une telle méthode repose sur la capacité à trouver une variable explicative du multiplicateur de Lagrange qui soit de taille restreinte. Nous montrons comment mettre en œuvre cette méthodologie pour des systèmes composés de sous-systèmes de petite taille ne s'influençant pas directement les uns et les autres, mais couplés entre eux par une contrainte globale, telle que la satisfaction d'un équilibre entre productions et demande.

Il reste que le type de modèle que nous étudions ici a une structure particulière, même s'il permet d'étudier une large classe de systèmes réels tels que le système de production électrique sur lequel nous nous attardons dans le chapitre 4. Dans un modèle plus général, les différents sous-systèmes peuvent non seulement contribuer à la satisfaction d'une contrainte globale, mais interagissent également "directement" entre eux, à la manière d'un réseau. La stratégie optimale de gestion d'un sous-système dépend alors directement de ce que va faire le voisin. On peut alors imaginer d'adapter la méthodologie décrite ici en dualisant les liens entre les différentes unités. Cette idée reste à étudier.

Chapitre 4

Résolution numérique d'un problème de commande optimale stochastique de grande taille

> L'esprit n'use de sa faculté créatrice
> que quand l'expérience lui en impose
> la nécessité.
>
> Henri Poincaré (1854-1912)

L'étude que nous présentons ici a été réalisée au sein du département d'Optimisation, Simulation, Risques et Statistique (OSIRIS) d'EDF R&D. Elle a grandement bénéficié des travaux des stages de Peio Lahirigoyen, alors étudiant à l'École Nationale Supérieure de Techniques Avancées (ENSTA) (voir Lahirigoyen, Barty, et Girardeau, 2008), de Basma Kharrat, alors étudiante à l'École Polytechnique (voir Kharrat, Barty, et Girardeau, 2009), et de Raphaël Glon, alors étudiant à l'ENSTA (voir Glon, Barty, et Girardeau, 2010), que j'ai eu la chance d'encadrer au cours de la thèse.

Nous illustrons l'étude menée au chapitre 3 par la mise en œuvre de la méthode de décomposition stochastique proposée dans le cadre de la gestion de la production d'un parc électrique. Il ne s'agit pas d'un problème nouveau : la commande de tels systèmes a fait l'objet d'importantes études, notamment à EDF (directement ou via des partenariats avec les universitaires), depuis les années 70 environ. La complexité de ce problème, sur laquelle nous reviendrons en détail, est telle qu'il n'est pas encore possible de le résoudre exactement de nos jours.

Nous disposons d'un certain nombre de réserves d'énergie (hydrauliques notamment) participant à la satisfaction d'une demande globale[1] en puissance à chaque instant, le restant étant assuré par des unités thermiques dont nous considérerons le stock comme "illimité". Contrairement aux premières, l'usage des moyens de production thermiques a un certain coût. Le système considéré est naturellement soumis à des aléas (demande en électricité, apports dans les réserves, pannes sur les centrales thermiques, etc.). L'objectif est de minimiser l'espérance du coût de production sur un horizon de temps donné.

La programmation dynamique est en principe la bonne méthode numérique permettant de trouver les stratégies de gestion optimales du système. Cependant, elle n'est plus applicable en pratique dès lors que le nombre de réserves dépasse quelques unités, même

1. On négligera dans ce modèle les problématiques de réseau.

si des développements récents (voir, notamment Vezolle *et al.*, 2009) tirent parti du calcul parallèle pour repousser quelque peu les limites communément admises. Pour autant, la programmation dynamique ne s'applique pas au problème étudié ici.

Comme le système est composé de sous-systèmes, que sont les unités de production, on souhaiterait naturellement décomposer le problème de contrôle optimal en plus petits problèmes. Mais le fait que chaque unité de production participe à la satisfaction d'une demande globale les couple toutes entre elles. Il est possible de relaxer ce couplage en considérant des stratégies de gestion qui pour chaque unité ne dépendent que de l'aléa local, le comportement des autres sous-systèmes étant connu seulement en moyenne ou en loi, comme le proposent Delebecque et Quadrat (1978). Cependant, vu le couplage inhérent au problème (et qui provient notamment du fait qu'il existe des aléas "globaux" tels que la demande), l'utilisation d'une telle méthodologie nécessiterait de recalculer les stratégies à chaque nouveau début de période (technique dite de "rebouclage" ou de "boucle ouverte adaptée"). Ce n'est donc pas adapté à la recherche de stratégies qui soient valables pour tout l'horizon de temps.

Turgeon (1980) propose une méthode de décomposition par agrégation qui consiste à rechercher les stratégies de gestion de chaque unité en *feedback* sur l'information locale de l'unité concernée et sur une information agrégée représentant le reste du parc. Ces modèles sont alors résolus par programmation dynamique, qui devient praticable car, du fait de l'agrégation, la variable d'état est de dimension 2. C'est cette méthode qui a été développée à EDF (voir Lederer, Torrion, et Bouttes, 1984, Torrion et Leveugle, 1985) et qui est encore en fonction aujourd'hui. Nous comparerons nos résultats à une méthode de ce type au cours du chapitre.

Beaucoup d'autres méthodologies ont bien sûr été proposées pour résoudre le problème de gestion de gestion du parc dans l'incertain. On peut chercher à approcher la fonction de Bellman sous une forme paramétrique à la manière de Gal (1989) avec la *Parameter Iteration Method*, qui a été appliquée à des problèmes de gestion de réserves en grande dimension. Pereira et Pinto (1991) proposent, à travers la méthode de *Stochastic Dual Dynamic Programming* (SDDP), d'approcher la fonction valeur du problème par l'enveloppe supérieure d'hyperplans affines, mise à jour de manière itérative. Cette méthode, dont on trouvera une preuve de convergence dans l'article de Philpott et Guan (2008) et une analyse fine dans le travail de Shapiro (2010), a rencontré un franc succès dans l'industrie, puisqu'elle fait partie des rares permettant d'obtenir des fonctions de Bellman globales et possédant des garanties de convergence, au moins dans le cas convexe. Par ailleurs, le mariage entre méthodes de décomposition-coordination et discrétisation sur un arbre de scénarios a fait l'objet de nombreuses études et applications (Carpentier, Cohen, Culioli, et Renaud, 1996, Bacaud *et al.*, 2001, Emiel et Sagastizábal, 2010).

4.1 Formulation du problème

Le problème général de gestion du portefeuille de production électrique, tel qu'il se pose à EDF ainsi que pour beaucoup d'autres producteurs, est un problème d'optimisation complexe pour plusieurs raisons :

Horizon de temps et discrétisation Le problème se pose généralement sur un horizon pluri-annuel, du fait que certaines décisions, telles que le rechargement en combustible des centrales nucléaires, doit être décidé plusieurs années à l'avance. Pour autant, la discrétisation en temps doit être suffisamment fine pour pouvoir

modéliser les phénomènes de pointe ("courts" instants, de l'ordre de l'heure, où la demande en électricité croît de manière importante), car c'est à ces moments que se crée une grande partie de la valeur du portefeuille. Cela implique a priori un modèle comportant plusieurs milliers de pas de temps.

Stochasticité À un tel horizon de temps, beaucoup de paramètres du problème sont aléatoires : la demande, les apports dans les réserves hydrauliques, le prix de l'électricité sur le marché, le prix des combustibles, les éventuelles pannes des centrales, etc. Ces variables aléatoires sont de plus observées au moment où elles se réalisent et peuvent alors être utilisées pour prendre les décisions futures, ce qui est bien sûr avantageux du point de vue économique. Mais cela complique le problème mathématique : on est dans le cas de la boucle fermée évoqué au chapitre 1.

Non-convexité Le fait, par exemple, de mettre en marche une centrale thermique nécessite d'utiliser une certaine quantité de combustible, qui est alors perdue, ce qui se traduit par un certain coût fixe de démarrage. Ainsi, la courbe production-coût des centrales thermiques est en général non-convexe. De plus, le lien entre turbiné et production électrique pour une centrale hydraulique fait également intervenir le stock du lac, ceci de manière non-convexe : c'est le phénomène physique de hauteur de chute qui fait que la puissance produite est d'autant plus importante qu'il y a d'eau au-dessus de la turbine.

Nombre de stocks Le nombre d'unités de production devant être prises en compte est de l'ordre de plusieurs dizaines. On pense bien sûr aux stocks de combustibles, dont les stocks hydrauliques et les contrats d'effacement, mais on doit en général considérer d'autres types de stock pour prendre en compte la durée minimum de marche et d'arrêt des centrales, les contraintes d'émission de polluants, etc. L'hétérogénéité des caractéristiques de ces stocks nous empêche de les agréger directement.

Toutes ces difficultés ne peuvent (pour l'instant ?) pas être traitées simultanément. Ainsi, l'approche qui est généralement choisie consiste à considérer plusieurs problèmes à différents horizons de temps :

1. À court terme, c'est-à-dire d'un jour à l'autre, on considère une discrétisation très fine en temps et une modélisation précise du parc (en particulier, on inclut toutes les non-convexités). On néglige en revanche la stochasticité des paramètres du problème. Ce problème est posé sur un horizon qui est de l'ordre de la journée, et il s'agit de trouver des programmes réalisables pour l'ensemble des moyens de production du parc. L'un des points critiques est de bien fixer la valeur de fin de jeu du problème : il s'agit de la valeur que l'on donne au stock restant à l'issue de la journée. Une valeur trop faible pousserait l'optimiseur à définir une stratégie qui vide les stocks en fin de journée. À l'inverse, une valeur trop forte nous ferait aboutir à une stratégie conservant le plus possible le stock en fin de journée. La valeur de fin de jeu est en pratique issue de la résolution du problème à moyen terme qui suit.
2. À moyen terme, c'est-à-dire deux à trois ans à l'avance, on considère une modélisation grossière du parc, mais on modélise plus finement les variables aléatoires du problème, telles que la demande, les pannes et les apports hydrauliques. L'objectif est d'estimer la fonction de fin de jeu du problème à court terme, mais aussi d'estimer un certain nombre d'indicateurs économiques à l'avance : besoins en combustibles, coûts des programmes, indicateurs de risque, etc. Bien que ce problème soit a priori en horizon infini avec une périodicité annuelle, on constate qu'il suffit de

considérer un horizon de trois ans environ pour que les lois des différentes quantités en présence se stabilisent.

3. À plus long terme, on s'intéresse au placement optimal des dates d'arrêts des centrales nucléaires. C'est un problème difficile par son caractère combinatoire et stochastique et important en terme économique du fait de la part importante de l'énergie nucléaire dans la production d'électricité en France. Nous n'insisterons pas ici sur ce problème. Il faut également inclure, à cet horizon de temps, les problématiques d'investissement optimal sur le parc de production.

Dans la suite de ce chapitre, nous nous intéressons à un problème moyen-terme, où nous cherchons à optimiser les stratégies de gestion des réserves sur un horizon de 163 semaines, soit un peu plus de trois ans. Nous considérons deux types de moyens de production d'énergie.

- D'une part, nous diposons d'unités de production thermiques de diverses sortes (fuel, charbon, gaz, nucléaire), ainsi que d'un marché sur lequel nous pouvons vendre ou acheter de l'énergie. Pour tous ces moyens de production, nous ne considérons pas de problématique de stock : nous supposerons qu'il est toujours possible pour ces unités de produire la puissance demandée (réserve "infinie"). À chacune de ces unités est associé un certain coût unitaire de production, qui est aléatoire du fait des variations de prix et des éventuelles pannes.
- D'autre part, nous disposons d'unités possédant un stock fini d'énergie, et pouvant éventuellement être réapprovisionnées au cours de la période. La production d'énergie à l'aide de ces unités n'a en général pas de coût direct [2]. Cependant, du fait de la rareté du stock et du coût des moyens de productions thermiques, le stock possède une certaine valeur indirecte. Comme il s'agit souvent de réserves hydrauliques, la valeur unitaire du stock est communément appelée valeur de l'eau en pratique. Ces réserves sont de deux types. En premier lieu, nous avons des lacs qui sont en réalité le résultat de l'agrégation d'un grand nombre de réserves hydrauliques. En second lieu, nous disposons de contrats dits d'effacement de jour de pointe (EJP) qui permettent d'inciter les clients ayant souscrit à l'offre EJP à réduire leur consommation d'énergie, ceci un certain nombre de jours dans l'année. Les jours en question sont décidés par le producteur et nous considérons donc ce type de contrat comme un stock d'énergie supplémentaire.

Dans le problème moyen-terme considéré, il s'agit de gérer de manière optimale l'usage de 7 réserves d'énergie (lacs, contrats d'effacement). À chaque instant la somme des productions de ces réserves et des moyens thermiques, au nombre de 122 (nucléaire, fuel, gaz, charbon, marché), doit égaler la demande globale en puissance. Les variables aléatoires influant sur le système sont :

- la demande en électricité à chaque instant ;
- les coûts et puissances de productions des unités thermiques et du marché ;
- les apports en énergie des réserves.

Remarque 4.1 (Agrégation des usines thermiques). À chaque instant, les unités thermiques et le marché n'ont pas de stock et sont seulement caractérisées par une puissance maximale et un coût unitaire. Comme il est toujours économiquement plus intéressant de commencer à produire via les unités les moins onéreuses, il nous suffit de considérer une seule variable réelle, la production totale thermique, à chaque pas de temps, et non 122 variables de production où chacune correspondrait à une des unités thermiques. Nous notons $\boldsymbol{U}_t^{\text{th}}$

2. Le coût de production hydraulique est considéré comme négligeable.

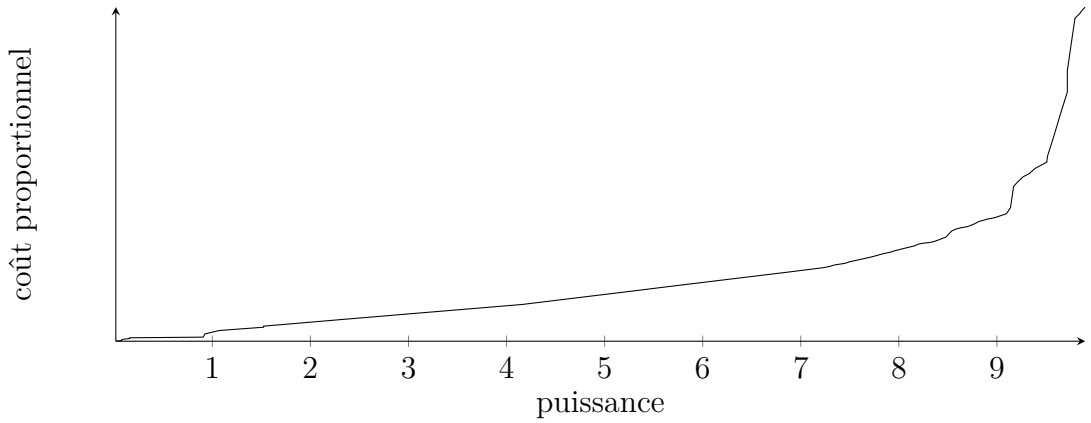

FIGURE 4.1 – Courbe de coût unitaire thermique

cette production totale thermique à l'instant t. Nous traçons la courbe de coût unitaire de production thermique en fonction de la production demandée dans la figure 4.1. On observe que cette courbe est strictement croissante, ce qui indique que sa primitive, la courbe de coût total thermique, est strictement convexe [3]. Cette propriété est nécessaire pour garantir que, du moins dans le cas déterministe, un algorithme de résolution tel que celui que nous décrivons par la suite converge vers la solution optimale du problème.

Notons I^{hy} l'ensemble des indices correspondant aux réserves et $T = 163$ l'horizon de temps du problème. Le problème se pose alors comme suit [4].

$$\min_{\boldsymbol{X},\boldsymbol{U}} \quad \mathbb{E}\left(\sum_{t=0}^{T-1} \boldsymbol{C}_t\left(\boldsymbol{U}_t^{\mathrm{th}}\right)\right), \tag{4.1a}$$

sous les conditions de dynamiques des stocks :

$$\boldsymbol{X}_0^i = x_0^i, \qquad \forall i \in I^{\mathrm{hy}}, \tag{4.1b}$$
$$\boldsymbol{X}_{t+1}^i = \boldsymbol{X}_t^i - \boldsymbol{U}_t^i + \boldsymbol{A}_t^i, \qquad \forall i \in I^{\mathrm{hy}}, \forall t = 0, \dots, T-1, \tag{4.1c}$$

la contrainte d'équilibre offre-demande :

$$\sum_{i \in I^{\mathrm{hy}}} \boldsymbol{U}_t^i + \boldsymbol{U}_t^{\mathrm{th}} = \boldsymbol{D}_t, \tag{4.1d}$$

les contraintes de borne sur l'état et sur la commande :

$$\underline{u}_t^{\mathrm{th}} \leq \boldsymbol{U}_t^{\mathrm{th}} \leq \overline{u}_t^{\mathrm{th}}, \qquad \forall t = 0, \dots, T-1, \tag{4.1e}$$
$$\underline{u}_t^i \leq \boldsymbol{U}_t^i \leq \overline{u}_t^i, \qquad \forall i \in I^{\mathrm{hy}}, \forall t = 0, \dots, T-1, \tag{4.1f}$$
$$\underline{x}_t^i \leq \boldsymbol{X}_t^i \leq \overline{x}_t^i, \qquad \forall i \in I^{\mathrm{hy}}, \forall t = 0, \dots, T, \tag{4.1g}$$

et la contrainte de non-anticipativité :

$$\boldsymbol{U}_t^i \preceq (\boldsymbol{W}_0, \dots, \boldsymbol{W}_t), \qquad \forall i \in I^{\mathrm{hy}}, \forall t = 0, \dots, T-1, \tag{4.1h}$$
$$\boldsymbol{U}_t^{\mathrm{th}} \preceq (\boldsymbol{W}_0, \dots, \boldsymbol{W}_t), \qquad \forall t = 0, \dots, T-1, \tag{4.1i}$$

3. Ce n'est pas le cas dans la réalité, où cette courbe est plutôt linéaire par morceaux.
4. Notons que la valeur de fin de jeu est nulle, pour les raisons qui ont été énoncées plus haut.

où on a noté $\boldsymbol{W}_t := (\boldsymbol{A}_t, \boldsymbol{C}_t, \boldsymbol{D}_t)$ l'ensemble des bruits affectant le système au pas de temps t. Précisons de plus que les contraintes (4.1c)–(4.1g) sont des contraintes $\mathbb{P}$-presque sûres.

On notera $\mathcal{A}_t$ la tribu engendrée par les variables aléatoires $\boldsymbol{W}_0, \ldots, \boldsymbol{W}_t$. Celle-ci représente l'information disponible à l'instant t pour prendre la décision. Le modèle de décision est ici du type "hasard-décision", c'est-à-dire que la décision à un certain instant est prise une fois que les aléas affectant le système à cet instant sont connus. Ce choix de modélisation provient du fait qu'en mode "décision-hasard" la contrainte d'équilibre offre-demande ne serait pas réalisable (on ne peut pas égaler production et demande sans connaître la valeur de cette dernière).

Pendant la phase d'optimisation, on supposera que les bruits sont indépendants pas de temps par pas de temps [5]. Dès lors, nous nous trouvons dans le cadre de la programmation dynamique et nous savons qu'il n'y a pas de perte d'optimalité à chercher la stratégie optimale à chaque instant t comme une fonction de la variable $\boldsymbol{X}_t$, que l'on appelle alors variable d'état, et du bruit à l'instant t (puisque nous nous sommes placés en hasard-décision).

Soit $V_t(x)$ le coût optimal du problème partant de l'instant t en l'état x et N le cardinal de l'ensemble I^{hy}. On a l'équation de programmation dynamique suivante [6].

$$V_T(x) = 0, \qquad \forall x \in \mathbb{R}^N,$$

et, pour chaque instant $t = 0, \ldots, T-1$:

$$V_t(x) = \mathbb{E}\left(\min_{u \in \mathbb{R}^N} \boldsymbol{C}_t \left(\boldsymbol{D}_t - \sum_{i=1}^{N} u^i \right) + V_{t+1}(x - u + \boldsymbol{A}_t) \right), \qquad \forall x \in \mathbb{R}^N.$$

Du fait de la relative grande dimension de l'état, il n'est pas possible de résoudre cette équation en pratique à l'aide d'une discrétisation brutale de l'espace d'état. C'est pourquoi nous nous tournons vers les méthodes de décomposition, et en particulier celle du chapitre 3, pour résoudre le problème.

4.2 De l'importance de la simulation

La résolution numérique du problème (4.1) doit permettre d'obtenir des stratégies de gestion utilisables en pratique, ce qui signifie qu'elles doivent pouvoir s'implanter dans un processus de décision réaliste. De plus, on doit pouvoir calculer le coût associé à ces stratégies afin de pouvoir, par exemple, classer des méthodes entre elles. En effet, dans la plupart des cas que nous rencontrons en commande optimale stochastique, il n'est pas possible de tester si l'on se trouve à l'optimum. De plus, il est en général impossible de calculer la valeur de la fonction objectif de manière exacte. C'est la raison pour laquelle il est important de mettre en œuvre plusieurs méthodes (ou plusieurs variantes d'une méthode) et de les comparer à l'aide d'un protocole d'évaluation indépendant du processus d'optimisation.

La bonne manière de répondre à ces deux obligations est de construire un outil appelé simulateur qui soit capable de modéliser le processus de décision réel. Il faut bien

5. Il serait tout à fait possible de considérer un modèle d'aléas plus complexes faisant intervenir de la mémoire sur la demande, par exemple. Cela se ferait cependant au prix d'augmenter la taille de l'état, ici la variable $\boldsymbol{X}$.

6. où l'on a omis les contraintes de bornes sur la commande comme sur l'état pour alléger l'écriture

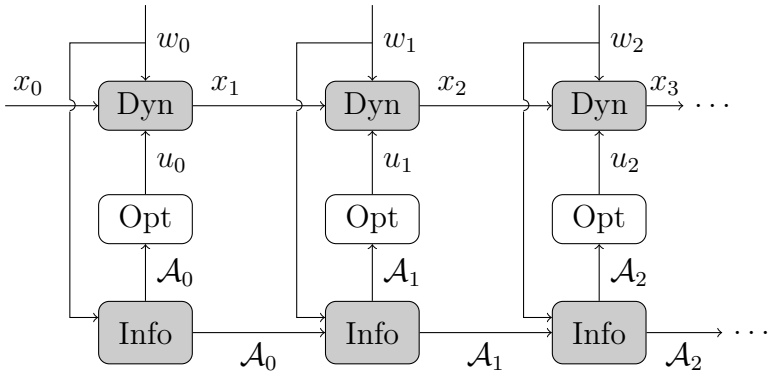

FIGURE 4.2 – Simulateur (Opt=Optimiseur, Dyn=Dynamiques, Info=Information)

distinguer modèle d'optimisation et modèle de simulation. Le modèle de simulation doit représenter le plus finement possible le processus réel et doit être unique. À l'inverse, chaque méthode de résolution du problème peut considérer son propre modèle en optimisation [7] afin de construire des stratégies. La seule obligation est que les stratégies doivent pouvoir s'implanter dans le simulateur. L'unicité du simulateur garantit que l'on compare des quantités comparables.

À chaque instant t, le simulateur dispose de toutes les observations passées, ici le passé du bruit $\boldsymbol{W}_0, \dots, \boldsymbol{W}_t$, qu'il transmet à l'optimiseur (décideur). Ce dernier doit rendre une commande $\boldsymbol{U}_t$ que le simulateur utilise afin de faire évoluer le système jusqu'à l'instant suivant et de calculer le coût associé à cette commande. Le processus continue ainsi jusqu'au bout de l'horizon. Un tel mécanisme est décrit dans la figure 4.2. Ce que nous cherchons à construire constitue la boîte blanche, ce qui ne signifie pas que tous les calculs relatifs à l'optimisation doivent être réalisés à ce moment dans le processus de simulation. Par exemple, dans le cas de la programmation dynamique, les fonctions de Bellman sont calculées en amont et les opérations faites dans la boîte blanche reviennent juste à une évaluation de ces fonctions de Bellman.

Ce que nous venons de décrire est la procédure de simulation sur une seule trajectoire de bruit. Par la suite, nous effectuerons cette prodécure sur un grand nombre de trajectoires indépendantes du bruit, afin de faire des estimations par Monte-Carlo.

Remarque 4.2. En pratique, le simulateur retient aussi à chaque instant des variables annexes utiles à la simulation, notamment la variable $\boldsymbol{X}_t$ qui représente ici le stock restant dans une réserve. Notons que ceci n'apporte pas d'information supplémentaire à $\boldsymbol{W}_0, \dots, \boldsymbol{W}_t$, puisque les variables que l'on retient à l'instant t sont $\mathcal{A}_t$-mesurables. À titre d'exemple, on fournit en pratique à l'optimiseur les variables $\boldsymbol{X}_t, \boldsymbol{W}_0, \dots, \boldsymbol{W}_t$.

Décrivons maintenant plus précisément la manière dont nous allons procéder. Pour toute stratégie $\boldsymbol{U} = (\boldsymbol{U}_0, \dots, \boldsymbol{U}_{T-1})$ et pour toute réalisation du processus de bruit $\boldsymbol{W} = (\boldsymbol{W}_0, \dots, \boldsymbol{W}_T)$, le simulateur permet d'évaluer le coût de gestion $j(\boldsymbol{U}, \boldsymbol{W})$. Comme nous l'avons déjà dit, nous ne savons pas calculer de manière exacte le critère $J(\boldsymbol{U}) := \mathbb{E}\left(j\left(\boldsymbol{U}, \boldsymbol{W}\right)\right)$. Afin de comparer les stratégies issues de différentes méthodes entre elles, disons $\boldsymbol{U}_1$ et $\boldsymbol{U}_2$, nous allons donc tester statistiquement l'hypothèse $J(\boldsymbol{U}_1) < J(\boldsymbol{U}_2)$. Pour ce faire, nous allons estimer, à l'aide d'échantillons du bruit $\boldsymbol{W}$, la distribution de

7. Par exemple, nous supposerons ici, lors de la phase d'optimisation, que les aléas sont indépendants pas de temps par pas de temps.

	Coût moyen	IC95%
Borne inf.	2.014	$0.9 \cdot 10^{-2}$
Borne sup.	2.406	$1.7 \cdot 10^{-2}$

TABLE 4.1 – Bornes pour le coût du problème

la variable aléatoire :

$$j\left(\boldsymbol{U}_1, \boldsymbol{W}\right) - j\left(\boldsymbol{U}_2, \boldsymbol{W}\right),$$

ou encore, plus simplement, estimer sa moyenne, sa variance, ou tout autre indicateur statistique utile. Un bon indicateur sera le niveau de probabilité p défini par :

$$\mathbb{P}\left(j\left(\boldsymbol{U}_1, \boldsymbol{W}\right) - j\left(\boldsymbol{U}_2, \boldsymbol{W}\right) > 0\right) = p.$$

Une valeur de p grande indiquera que $\boldsymbol{U}_2$ constitue une meilleure stratégie que $\boldsymbol{U}_1$.

4.3 Méthode de référence

Afin d'avoir des éléments de comparaison, nous calculons des bornes supérieures et inférieures pour l'espérance du coût de gestion. Nous mettons également en œuvre une méthode alternative à DADP : la décomposition par agrégation.

4.3.1 Bornes supérieures et inférieures

Pour obtenir une borne supérieure, nous mettons en œuvre au sein du simulateur une stratégie réalisable mais très sous-optimale. Celle-ci consiste à se servir des réserves à puissance maximale tant que cela est possible, c'est-à-dire tant que la réserve n'est pas vide. Il s'agit d'une stratégie qui oublierait de tenir compte des arbitrages temporels possibles.

La borne inférieure, quant à elle, est calculée en calculant le coût du problème où l'on a relâché les contraintes de bornes sur les stocks. L'optimum consiste alors à produire avec les réserves sans arrêt à puissance maximale, puis à compléter avec la production thermique, du moins coûteux au plus coûteux, jusqu'à atteindre la demande. Cette borne est probablement très éloignée du coût optimal, mais nous donne tout de même une idée sur ce que nous pouvons espérer gagner via la recherche de stratégies optimales.

Les résultats sont présentés dans le tableau 4.1. La colonne IC95% nous indique que le coût exact a 95% de chances de se trouver dans l'intervalle [Coût moyen ± IC95%]. Les résultats nous indiquent que le coût optimal du problème se situe approximativement entre 2 et 2.4.

4.3.2 Décomposition par agrégation

L'algorithme de décomposition par agrégation propose de remplacer un problème comportant n stocks par n sous-problèmes comportant chacun 2 stocks. Dans le sous-problème i, le preier stock correspond exactement au stock i, tandis que le second résulte

d'une agrégation des autres stocks. Chaque sous-problème étant de dimension 2, il est possible de lui appliquer la programmation dynamique. Dans la phase de simulation, on utilise chaque fonction de Bellman pour en déduire les décisions de l'unité correspondante. Cette méthodologie suppose que les structures, notamment les dynamiques, des réserves sont proches.

Nous présentons rapidement le sous-problème approché lors d'une décomposition par agrégation pour l'unité $i \in I^{\mathrm{hy}}$. On écrit un problème de gestion où la modélisation de la réserve i est inchangée mais les réserves restantes sont agrégées en une seule, que nous notons ici i^{c}. Le sous-problème agrégé associé à l'unité i s'écrit :

$$\min_{\boldsymbol{X}^i, \boldsymbol{U}^i, \boldsymbol{X}^{i^{\mathrm{c}}}, \boldsymbol{U}^{i^{\mathrm{c}}}} \quad \mathbb{E}\left(\sum_{t=0}^{T-1} \boldsymbol{C}_t\left(\boldsymbol{U}_t^{\mathrm{th}}\right)\right),$$

sous les conditions de dynamiques des stocks :

$$\begin{aligned}
&\boldsymbol{X}_0^i = x_0^i, \qquad \boldsymbol{X}_0^{i^{\mathrm{c}}} = x_0^{i^{\mathrm{c}}}, \\
&\boldsymbol{X}_{t+1}^i = \boldsymbol{X}_t^i - \boldsymbol{U}_t^i + \boldsymbol{A}_t^i, \qquad \forall t = 0, \dots, T-1, \\
&\boldsymbol{X}_{t+1}^{i^{\mathrm{c}}} = \boldsymbol{X}_t^{i^{\mathrm{c}}} - \boldsymbol{U}_t^{i^{\mathrm{c}}} + \boldsymbol{A}_t^{i^{\mathrm{c}}}, \qquad \forall t = 0, \dots, T-1,
\end{aligned}$$

la contrainte d'équilibre offre-demande :

$$\boldsymbol{U}_t^i + \boldsymbol{U}_t^{i^{\mathrm{c}}} + \boldsymbol{U}_t^{\mathrm{th}} = \boldsymbol{D}_t,$$

les contraintes de borne sur l'état et sur la commande :

$$\begin{aligned}
&\underline{u}_t^i \leq \boldsymbol{U}_t^i \leq \overline{u}_t^i, \qquad \underline{u}_t^{i^{\mathrm{c}}} \leq \boldsymbol{U}_t^{i^{\mathrm{c}}} \leq \overline{u}_t^{i^{\mathrm{c}}}, \qquad \forall t = 0, \dots, T-1, \\
&\underline{x}_t^i \leq \boldsymbol{X}_t^i \leq \overline{x}_t^i, \qquad \underline{x}_t^{i^{\mathrm{c}}} \leq \boldsymbol{X}_t^{i^{\mathrm{c}}} \leq \overline{x}_t^{i^{\mathrm{c}}}, \qquad \forall t = 0, \dots, T-1, \\
&\underline{u}_t^{\mathrm{th}} \leq \boldsymbol{U}_t^{\mathrm{th}} \leq \overline{u}_t^{\mathrm{th}}, \qquad \forall t = 0, \dots, T-1,
\end{aligned}$$

et les contraintes de non-anticipativité :

$$\boldsymbol{U}_t^i, \boldsymbol{U}_t^{i^{\mathrm{c}}} \preceq \boldsymbol{W}_0, \dots, \boldsymbol{W}_t, \qquad \forall i \in I^{\mathrm{hy}} \cup I^{\mathrm{th}}, \forall t = 0, \dots, T-1.$$

La conception des paramètres associés à la réserve agrégée i^{c}, tels que les apports, les bornes sur le stock et sur la commande, est généralement heuristique et dépend du problème traité. Il est clair que plus les réserves ont des caractéristiques proches, moins cette approximation sera forte. Au contraire, il paraît difficile d'appliquer cette méthodologie à des stocks très hétérogènes, parmi lesquels on compterait, par exemple, à la fois des stocks de combustibles, des stocks de polluants, des stocks d'heures, etc.

Sur le sous-problème i que nous venons d'expliciter, en faisant l'hypothèse que les bruits sont indépendants pas de temps par pas de temps, on peut effectuer une programmation dynamique sur grille. On obtient ainsi une fonction de Bellman de dimension 2, qui dépend du stock local et du stock agrégé. Celle-ci est implantable dans le simulateur et on peut donc simuler les stratégies obtenues par décomposition-agrégation afin de les comparer à l'algorithme DADP.

4.4 Application de DADP

On propose de résoudre de manière approchée le problème dual associé au problème primal (4.1) dans lequel on a dualisé la contrainte (4.1d) en utilisant l'algorithme DADP décrit au chapitre 3. La méthode mise en œuvre est basée sur une décomposition par les prix, adaptée au cas stochastique. À chaque itération, elle se découpe naturellement en deux étapes : la résolution des sous-problèmes à multiplicateur fixé, puis la mise à jour du multiplicateur.

On notera $\boldsymbol{\lambda}^k$ ce multiplicateur, dont il existe une version $\mathcal{A}_t$-mesurable, à l'itération k. Nous n'avons en général pas d'autre information sur sa dynamique ou sur ses corrélations avec les autres aléas. Nous en avons seulement des scénarios à chaque itération.

4.4.1 Sous-problème thermique

Le premier type de sous-problème, qui s'écrit :

$$\min_{\boldsymbol{U}^{\mathrm{th}}} \quad \mathbb{E}\left(\sum_{t=0}^{T-1}\left(\boldsymbol{C}_t\left(\boldsymbol{U}_t^{\mathrm{th}}\right)-\boldsymbol{\lambda}_t\boldsymbol{U}_t^{\mathrm{th}}\right)\right), \tag{4.2a}$$

sous la condition de mesurabilité :

$$\boldsymbol{U}_t^{\mathrm{th}} \preceq \boldsymbol{W}_0,\dots,\boldsymbol{W}_t, \qquad \forall t=0,\dots,T-1, \tag{4.2b}$$

et les contraintes de borne sur la commande :

$$\underline{u}_t^{\mathrm{th}} \le \boldsymbol{U}_t^{\mathrm{th}} \le \overline{u}_t^{\mathrm{th}}, \qquad \forall t=0,\dots,T-1, \tag{4.2c}$$

est parfaitement décomposable pas de temps par pas de temps et ne pose pas de problème particulier. Il s'agit en effet simplement de rechercher sur la courbe de la figure 4.1 l'abscisse du point ayant pour ordonnée $\boldsymbol{\lambda}_t$, puis à projeter sur la contrainte de borne (4.2c). On obtient ainsi une commande thermique en feedback directement sur le prix. Nous expliquons dans §4.5 comment obtenir une stratégie implantable dans le simulateur.

4.4.2 Sous-problème hydraulique

Le deuxième type de sous-problème, qui concerne les réserves, s'écrit, pour chaque $i \in I^{\mathrm{hy}}$:

$$\min_{\boldsymbol{X}^i,\boldsymbol{U}^i} \quad \mathbb{E}\left(\sum_{t=0}^{T-1}-\boldsymbol{\lambda}_t\boldsymbol{U}_t^i\right), \tag{4.3a}$$

sous les conditions de dynamiques des stocks :

$$\boldsymbol{X}_0^i = x_0^i, \tag{4.3b}$$

$$\boldsymbol{X}_{t+1}^i = \boldsymbol{X}_t^i - \boldsymbol{U}_t^i + \boldsymbol{A}_t^i, \qquad \forall t=0,\dots,T-1, \tag{4.3c}$$

la contrainte de non-anticipativité :

$$\boldsymbol{U}_t^i \preceq \boldsymbol{W}_0,\dots,\boldsymbol{W}_t, \qquad \forall t=0,\dots,T-1, \tag{4.3d}$$

	$\boldsymbol{Y}_t$	$\boldsymbol{U}_t^i \preceq$	V_t^i dépend de
Exp. 0	0	$\boldsymbol{X}_t^i, \boldsymbol{A}_t^i$	$\boldsymbol{X}_t^i$
Exp. 1	$\boldsymbol{D}_t$	$\boldsymbol{X}_t^i, \boldsymbol{A}_t^i, \boldsymbol{D}_t$	$\boldsymbol{X}_t^i$
Exp. 2	$\boldsymbol{D}_t, \overline{\boldsymbol{P}}_t$	$\boldsymbol{X}_t^i, \boldsymbol{A}_t^i, \boldsymbol{D}_t, \overline{\boldsymbol{P}}_t$	$\boldsymbol{X}_t^i$

TABLE 4.2 – Trois choix pour la variable explicative du prix

et les contraintes de bornes :

$$\underline{u}_t^i \leq \boldsymbol{U}_t^i \leq \overline{u}_t^i, \qquad \forall t = 0, \dots, T-1, \tag{4.3e}$$

$$\underline{x}_t^i \leq \boldsymbol{X}_t^i \leq \overline{x}_t^i, \qquad \forall t = 0, \dots, T. \tag{4.3f}$$

Ce problème ne peut être découplé en temps du fait des contraintes de dynamique. De plus, le fait que nous n'ayons pas à notre disposition de dynamique sur le prix nous empêche d'écrire une équation de programmation dynamique en dimension raisonnable [8].

Nous suivons alors la méthodologie proposée dans le chapitre 3. Nous choisissons une variable $\boldsymbol{Y}_t$, dite variable d'information, qui doit être $\mathcal{A}_t$-mesurable et qui ne doit pas être influencée par les décisions. Nous considérons alors un nouveau sous-problème, du type :

$$\min_{\boldsymbol{X}^i, \boldsymbol{U}^i} \quad \mathbb{E}\left(\sum_{t=0}^{T-1} -\mathbb{E}\left(\boldsymbol{\lambda}_t \mid \boldsymbol{Y}_t\right) \boldsymbol{U}_t^i\right),$$

sous les mêmes contraintes de non-anticipativité, de dynamique des stocks, et de borne qu'auparavant. Nous allons considérer pour la variable [9] $\boldsymbol{Y}_t$ les trois choix présentés dans le tableau 4.2. Nous aurons pour chacun de ces "remplacements" des estimateurs statistiques (la déviance, notamment) qui nous indiqueront la pertinence de tel ou tel choix d'information en comparant l'espérance conditionnelle servant dans le sous-problème avec les scénarios de prix donnés par le coordonnateur.

On donne maintenant à titre d'exemple l'équation de programmation dynamique résolue pour le choix 1 : on se donne comme variable explicative pour le prix $\boldsymbol{Y}_t = \boldsymbol{D}_t$ et on obtient ainsi une commande qui, pour chaque unité i et à chaque instant t, est en *feedback* sur $(\boldsymbol{X}_t^i, \boldsymbol{A}_t^i, \boldsymbol{D}_t)$. L'idée derrière ce choix de variable explicative est de rajouter une information globale sur le système (la demande) dans la décision locale. La fonction de Bellman associée au sous-système i satisfait alors l'équation de programmation dynamique :

$$V_T^i(x) = 0, \qquad \forall x \in \mathbb{R},$$

et, pour chaque pas de temps $t = 0, \dots, T-1$:

$$V_t^i(x) = \mathbb{E}\left(\min_{u \in \mathbb{R}} -\mathbb{E}\left(\boldsymbol{\lambda} \mid \boldsymbol{D}_t\right) u + V_{t+1}^i\left(x - u + \boldsymbol{A}_t^i\right)\right), \qquad \forall x \in \mathbb{R}.$$

8. On peut en effet toujours se placer en feedback sur tout le passé du bruit, mais l'équation de programmation dynamique qui en découle n'est pas soluble en pratique.

9. que nous appellerons parfois variable explicative du prix

Il est à noter que l'espérance à chaque étape de la programmation dynamique porte uniquement sur $\boldsymbol{D}_t$ et $\boldsymbol{A}_t^i$. De plus, l'espérance conditionnelle peut tout à fait être calculée hors ligne, une fois par itération de l'algorithme et stockée sous la forme d'une fonction de la demande à l'instant t. Remarquons que la stratégie ainsi obtenue est directement implantable dans le simulateur. Elle dépend, à chaque instant, de la réalisation de la demande, de l'apport, ainsi que du stock local.

Remarque 4.3 (Critère linéaire). Le critère du sous-problème hydraulique est linéaire. Cela pose généralement des problèmes aux algorithmes basés, comme ici, sur une décomposition lagrangienne. Ce n'est cependant pas ce que nous constatons en pratique, probablement du fait que le critère du sous-problème thermique (la fonction $\boldsymbol{C}_t$) est lui fortement convexe. Nous avons observé que le comportement de l'algorithme est fortement dégradé lorsque nous considérons un critère pour le sous-problème thermique qui est seulement linéaire par morceaux. Nous énonçons au §4.4.3 des alternatives algorithmiques dans ce cas.

4.4.3 Coordination

La coordination se fait par une formule du type "gradient". Ainsi, à l'itération k, étant donné un multiplicateur $\boldsymbol{\lambda}^k$ et des stratégies $\boldsymbol{U}^{\sharp k}$ optimales vis-à-vis de ce multiplicateur, on met à jour le prix "scénario par scénario" :

$$\boldsymbol{\lambda}_t^{k+1} = \boldsymbol{\lambda}_t^k + \rho \left(\boldsymbol{D}_t - \sum_{i \in I^{\mathrm{hy}} \cup I^{\mathrm{th}}} \boldsymbol{U}_t^{i,\sharp k} \right).$$

Ce type du mise à jour suffit dans le cas où la fonction duale a des propriétés de régularité suffisantes, ce qui est garanti dans le cas où la fonction objectif du problème de départ est fortement convexe et où les contraintes définissent un ensemble réalisable convexe. Lorsque, par exemple, la fonction objectif est simplement convexe, on peut alors utiliser d'autres types de mises à jour pour le prix, tels que la méthode des faisceaux (voir Bonnans, Gilbert, Lemaréchal, et Sagastizábal, 1997). Une autre solution est de changer les sous-problèmes en leur ajoutant un terme quadratique incitant à mieux satisfaire la contrainte couplante, à la manière du Lagrangien Augmenté Linéarisé (voir Cohen et Zhu, 1984, Cohen, 1984). Outre le fait que lorsque le problème est non-convexe, la technique du Lagrangien Augmenté permet de réduire le saut de dualité, on a que son usage permet dans le cas simplement convexe de régulariser la fonction duale, ce qui a bien sûr un effet bénéfique sur le comportement de la méthode du gradient.

4.4.4 Décomposition par prédiction

Il existe d'autres manières d'effectuer l'étape de coordination. Au lieu d'utiliser une formule de gradient comme au §4.4.3, on peut utiliser la méthode par prédiction. Celle-ci nécessite qu'un des sous-systèmes soit capable de satisfaire à lui seul la contrainte couplante. Il s'agit ici du sous-problème thermique.

Supposons que l'on ait un processus de prix $\boldsymbol{\lambda}^k$ à l'itération k. Les sous-problèmes hydrauliques sont résolus comme expliqué au §4.4.2. Puis, le sous-problème thermique consiste simplement à satisfaire ce qui reste de la demande lorsqu'on lui a ôté la production des usines hydrauliques. De ce problème thermique, on sort le coût marginal de production, qui sert de nouveau processus de prix $\boldsymbol{\lambda}^{k+1}$.

La méthode par prédiction peut s'interpréter comme un algorithme de point fixe. Sa convergence ne repose donc pas sur le même type d'hypothèses que la décomposition par

les prix. On pourra consulter le cours de Cohen (2004, §3.3.3) pour de plus amples détails sur la décomposition par prédiction. Faute de temps, nous n'avons pas testé la mise en œuvre de cette méthode sur notre problème.

4.5 Résultats

On présente ici les performances des différentes variantes de décomposition stochastique qui ont été testées sur ce problème. Sur ce problème dont l'état est de dimension 7, nous ne sommes pas capables de mettre en œuvre une résolution de l'équation de programmation dynamique par discrétisation de l'espace d'état. Ainsi, afin de se donner une idée de la qualité des solutions issues de la décomposition, nous avons commencé par chercher des bornes inférieure et supérieure pour le problème. Un stage mené à OSIRIS (Glon *et al.*, 2010) a permis de mettre en œuvre, sur le même simulateur, la méthodologie utilisée en opérationnel, basée sur l'idée de décomposition par agrégation de Turgeon (1980). Nous comparons donc la performance de cette méthode avec les différentes variantes de DADP.

Tous les coûts présentés ici proviennent du simulateur, qui calcule ceux-ci en mettant en œuvre les stratégies sur 484 chroniques "historiques" disponibles à EDF et censées être représentatives des aléas. On calcule sur ces 484 valeurs la moyenne empirique du coût et un intervalle de confiance [10].

4.5.1 Considérations pratiques sur DADP

Nous expliquons ici certains choix faits lors de la mise en œuvre de l'algorithme DADP. Deux points d'ordre numérique méritent particulièrement notre attention.

Estimation des espérances conditionnelles

Nous avons choisi d'estimer les espérances conditionnelles intervenant dans les sous-problèmes à l'aide de modèles additifs généralisés (GAM), proposés par Hastie et Tibshirani (1990). Constatant que l'estimation de l'espérance conditionnelle d'une variable aléatoire $\boldsymbol{Z}$ par rapport à des variables explicatives $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$ se heurte, comme pour la programmation dynamique, à la malédiction de la dimension, ces méthodes cherchent à faire pour le mieux dans le cadre de modèles additifs. L'estimation est de la forme :

$$\mathbb{E}\left(\boldsymbol{Z} \mid \boldsymbol{P}_1, \ldots, \boldsymbol{P}_n\right) \simeq \sum_{i=1}^{n} f_i\left(\boldsymbol{P}_i\right).$$

Les fonctions f_i sont des splines (polynômes par morceaux) dont les caractéristiques sont optimisées par validation croisée sur les données statistiques d'entrée. On trouvera une explication détaillée sur ce type de modèles et sa mise en œuvre dans le livre de Wood (2006). Il existe au sein du logiciel libre de statiques R (R Development Core Team, 2009) une boîte à outils qui se charge d'optimiser ces paramètres. Elle donne également en sortie des indicateurs de qualité sur l'estimation. On utilisera en particulier l'indicateur de déviance, qui vaut 0 lorsque $\boldsymbol{Z}$ est estimé par son espérance $\mathbb{E}\left(\boldsymbol{Z}\right)$ et vaut 1 si notre estimateur est exact, autrement dit si $\sum_{i=1}^{n} f_i(\boldsymbol{P}_i) = \boldsymbol{Z}$.

10. Dans toute cette partie, on ne donne volontairement pas l'ordre de grandeur des coûts et des puissances en jeu pour des questions de confidentialité.

Remarque 4.4 (Estimateur à noyaux). Le choix de modèles additifs généralisés pour estimer les espérances conditionnelles a été fait suite à une comparaison avec les méthodes à noyaux (Nadaraya, 1964, Watson, 1964) effectuée au sein du logiciel R. Bien que les deux aient eu des performances comparables, les modèles additifs généralisés se sont avérés plusieurs dizaines de fois plus rapides que les méthodes à noyaux, d'où le choix fait.

Obtention de stratégies réalisables

Comme il a été précisé au §3.2.2, les stratégies obtenues par l'algorithme DADP ne satisfont pas a priori la contrainte d'équilibre production-demande, pas plus que dans le cas de la méthode de décomposition par agrégation. Cependant, pour le problème qui nous intéresse, nous sommes capables de construire, en phase de simulation des stratégies réalisables en jouant sur la production thermique. Expliquons de quelle manière.

Comme nous l'avons déjà constaté plus haut, la difficulté principale est l'obtention de stratégies de gestion des réserves. La production thermique n'étant pas, dans notre modèle, soumise à des contraintes de stock, elle est bien plus facile à placer. Ainsi, en phase de simulation, nous simulons les trajectoires des réserves à l'aide des stratégies issues de l'optimisation mais nous n'utilisons pas les stratégies issues de l'optimisation pour gérer le thermique : nous faisons en sorte que la production thermique vienne compléter exactement la production issue des stocks, de telle sorte que l'équilibre offre-demande soit assuré. Pour notre modèle, cela est toujours possible.

Dans le contexte de la décomposition, une telle procédure pourrait être appliquée au cours des itérations de l'algorithme. On pourrait alors calculer des scénarios de coûts marginaux du système en garantissant l'équilibre offre-demande à chaque itération. Si on change alors la règle de mise à jour des multiplicateurs énoncée au 4.4.3 en utilisant ensuite ces coûts marginaux comme les nouveaux prix, on retrouve le schéma de décomposition par prédiction dont nous avons parlé au §4.4.4.

4.5.2 Comparaison

On met en œuvre trois variantes de la méthode de décomposition stochastique qui correspondent aux trois qualités différentes d'estimation de l'espérance conditionnelle du prix au sein des sous-problèmes présentées dans le tableau 4.2.

- Dans la première expérience, on remplace le prix à chaque pas de temps par son espérance. Autrement dit, on explique le prix uniquement par la variable "temps". On peut alors résoudre chaque sous-problème i par programmation dynamique en dimension 1 (le stock de l'unité i) et on obtient des stratégies qui, pour chaque unité i et à chaque instant t, dépendent de $\boldsymbol{X}_t^i$ et de $\boldsymbol{A}_t^i$.
- Dans la deuxième expérience, on remplace le prix à chaque pas de temps par son espérance conditionnellement à la demande. Autrement dit, on explique le prix via la variable "temps" et la variable "demande". On a toujours à résoudre, pour chaque unité, une équation de programmation dynamique en dimension 1. On obtient maintenant, pour chaque unité i et pour chaque instant t, une stratégie dépendant de $\boldsymbol{X}_t^i$, $\boldsymbol{A}_t^i$ et $\boldsymbol{D}_t$.
- Dans la troisième expérience, on remplace le prix à chaque pas de temps par son espérance conditionnellement à la demande et à la disponibilité thermique[11]. Autrement dit, on explique le prix via la variable "temps", la variable "demande" et

11. Il s'agit d'un scalaire obtenu directement à partir de la courbe de coût thermique $\boldsymbol{C}_t$ qui varie selon la température et les pannes des différentes centrales thermiques, permettant ainsi d'indiquer le caractère

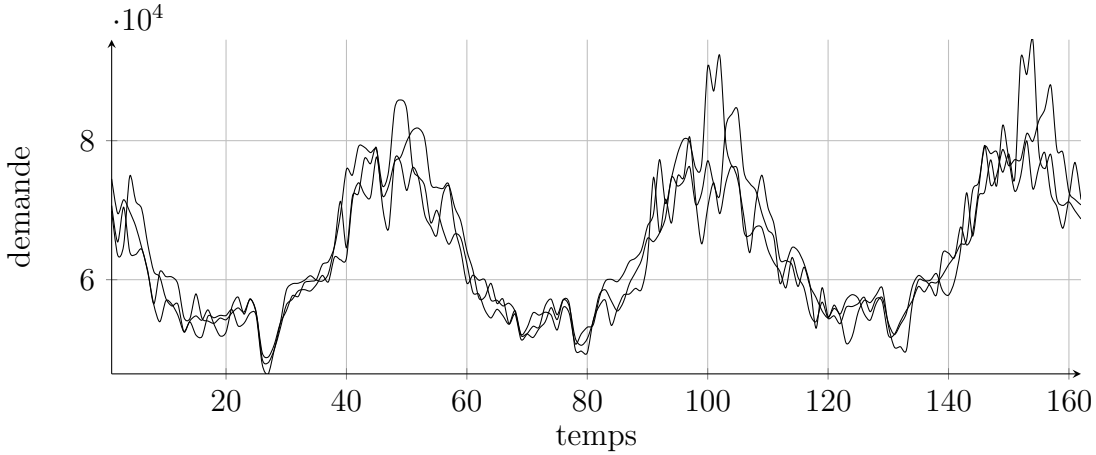

FIGURE 4.3 – Quelques scénarios de demande en puissance

	Coût moyen	IC95%	Déviance EC
Exp. 0	2.363	$1.3 \cdot 10^{-2}$	50.0%
Exp. 1	2.340	$1.3 \cdot 10^{-2}$	82.4%
Exp. 2	2.338	$1.3 \cdot 10^{-2}$	86.1%

TABLE 4.3 – Résultats de la décomposition stochastique

la variable "disponibilité thermique" et on obtient une stratégie qui, pour chaque unité i et pour chaque instant t, dépend de $\boldsymbol{X}_t^i$, $\boldsymbol{A}_t^i$, $\boldsymbol{D}_t$ et $\overline{\boldsymbol{P}}_t$.

On montre sur la figure 4.3 quelques scénarios de la demande en puissance, à titre d'exemple du type d'aléa affectant le système. On observe clairement la saisonnalité annuelle qui traduit le fait que la demande en électricité est nettement plus forte en hiver qu'en été. Naturellement, nous retrouverons cette forme sur les scénarios de coûts marginaux calculés par DADP. Ayant considéré un pas de temps hebdomadaire, nous ne pouvons observer les deux autres saisonnalités importantes de la demande "en temps continu" qui sont les saisonnalités journalière et hebdomadaire.

Les résultats obtenus par DADP au bout de 10 itérations sont présentés dans le tableau 4.3. La colonne "Déviance" indique la qualité de l'estimation du prix à l'aide des variables explicatives choisies. Comme on l'a déjà expliqué, plus la déviance est grande, mieux le prix est expliqué. Cet indicateur est calculé directement par l'outil qui opère la régression par GAM.

Remarque 4.5 (Nombre d'itérations). Nous sommes bien conscients que 10 itérations d'un algorithme de dual ne sont en général pas suffisantes pour espérer être proche de l'optimum. Sur le problème qui nous intéresse, il semble pour autant que le comportement de la méthode soit stabilisé à l'issue de 10 itérations. Il faut noter que les intervalles de confiance que nous obtenons sont trop grands pour nous permettre d'observer une amé-

tendu ou non de l'état du parc de production à chaque instant et pour chaque scénario.

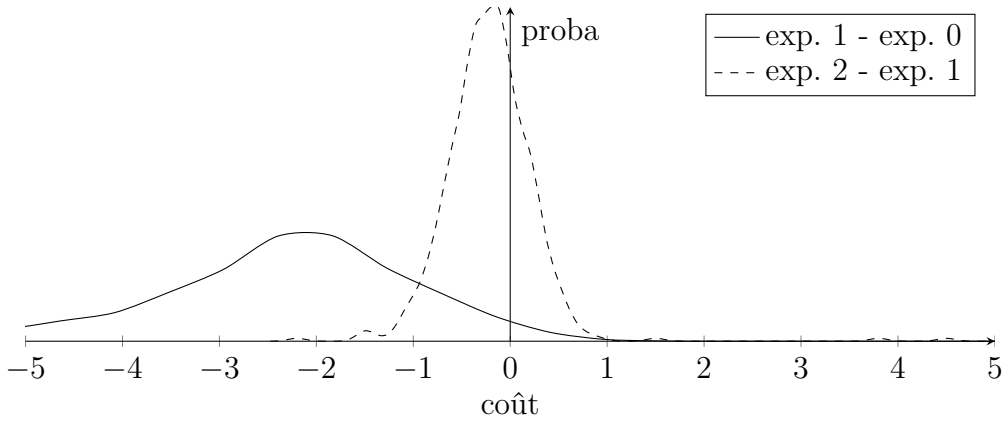

FIGURE 4.4 – Distribution de la différence des coûts entre les variantes de DADP

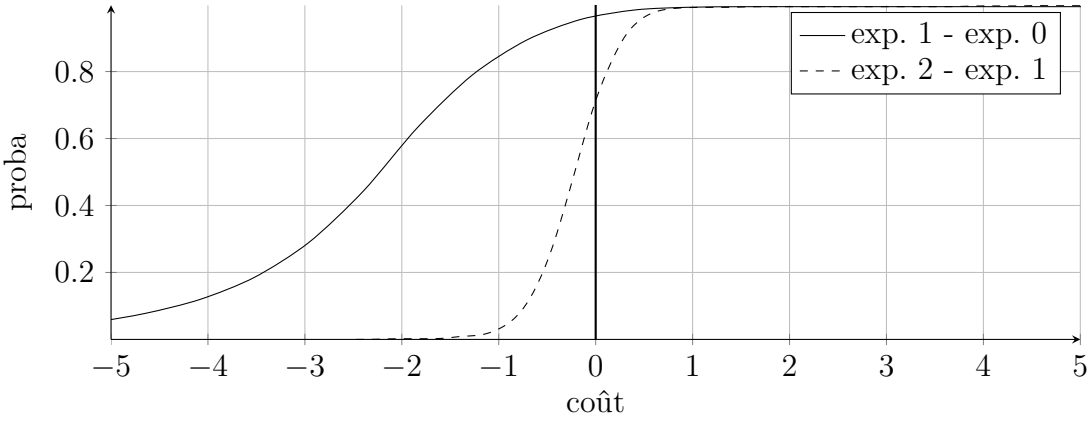

FIGURE 4.5 – Fonction de répartition de la différence des coûts entre les variantes de DADP

lioration du coût dès lors que l'on dépasse quelques itérations. De plus, comme nous le verrons au cours de la comparaison entre les différentes stratégies obtenues, il semble que ce problème ait la propriété d'avoir un minimum assez plat : il n'est pas très difficile d'obtenir une stratégie "raisonnable" offrant un coût proche de l'optimum.

On observe bien la diminution du coût optimal avec le raffinement de la méthode, ainsi que l'augmentation de la déviance. Cela dit, les deux dernières expériences sont en fait quasi-indistingables. C'est la raison pour laquelle nous traçons figure 4.4 la distribution de la différence des coûts obtenus en simulation entre l'expérience 0 et l'expérience 1, d'une part, et la différence des coûts entre l'expérience 1 et l'expérience 2, d'autre part. Nous traçons les fonctions de répartition correspondantes figure 4.5. Ainsi, autant il est clair que la stratégie de l'expérience 1 obtient un coût moyen meilleur que celle de l'expérience 0, puisqu'on observe que :

$$\mathbb{P}\left(j\left(\boldsymbol{U}^{\text{Exp. }1}, \boldsymbol{W}\right) - j\left(\boldsymbol{U}^{\text{Exp. }0}, \boldsymbol{W}\right) < 0\right) \simeq 0,95,$$

autant il est difficile de départager les stratégies de l'expérience 1 et de l'expérience 2, puisqu'on observe que :

$$\mathbb{P}\left(j\left(\boldsymbol{U}^{\text{Exp. }2}, \boldsymbol{W}\right) - j\left(\boldsymbol{U}^{\text{Exp. }1}, \boldsymbol{W}\right) < 0\right) \simeq 0,7.$$

Nous traçons dans la figure 4.6 l'évolution des valeurs primales et duales au cours des itérations, estimées sur l'échantillon qui nous sert à effectuer les mises à jour du multiplicateur de Lagrange. À chaque itération, comme nous l'avons expliqué, nous simulons les stratégies courantes afin de calculer l'écart entre production et demande et de mettre à jour le multiplicateur de Lagrange. Nous en profitons pour calculer, sur cet échantillon, une estimation de la valeur du problème dual et de la valeur du problème primal (où la commande est rendue réalisable par la technique expliquée au §4.5.1). Nous observons figure 4.6 que le coût dual croît au cours des itérations (ce qui est rassurant du fait que nous mettons en œuvre un algorithme de gradient sur le problème dual, qui est concave), et que le coût primal décroît. Au bout de quelques itérations, ces valeurs se stabilisent, notamment la valeur primale. Il subsiste cependant un écart, dont au moins une partie provient du fait que nous ne cherchons la solution du problème dual que dans un sous-espace de l'espace dual, ce qui nous empêche de converger vers le multiplicateur optimal.

Nous vérifions ensuite, dans la figure 4.7, l'interprétation faite au §3.2.2 de notre projection du multiplicateur de Lagrange sur la satisfaction dela contrainte. Nous avions alors montré, à l'aide de la théorie de la dualité, que la contrainte que nous dualisons ne serait satisfaite, à l'issue de la mise en œuvre de l'algorithme, que conditionnellement à la variable d'information choisie. Dans notre cas, nous devons donc observer, au cours des itérations, qu'au moins $\mathbb{E}\left(\boldsymbol{D}_t - \sum_{i=1}^{n} \boldsymbol{U}_t^i\right)$ tend, à chaque instant t, vers 0. C'est en effet ce que nous observons sur la figure 4.7. Pour l'expérience 1 (resp. 2), on a même plus : c'est l'espérance de l'écart conditionnellement à la demande (resp. conditionnellement à la demande et à l'indisponibilité thermique) qui doit tendre vers 0 au cours des itérations. Du fait de ces conditionnements, il est plus difficile d'observer cette décroissance graphiquement.

Plus précisément, nous traçons dans la figure 4.8 la distribution de cette écart entre production et demande, pour un pas de temps donné, au cours des itérations. On observe bien que la distribution de l'écart se déplace pour se centrer autour de 0 au fur et à mesure des itérations.

Intéressons-nous maintenant aux valeurs de multiplicateurs obtenus à l'issue de la mise en œuvre de l'algorithme. Nous traçons dans la figure 4.9 quelques scénarios de coûts marginaux tels qu'ils sont calculés par DADP dans le cas de l'expérience 2. On retrouve la saisonnalité annuelle qui fait que le coût marginal est plus élevé en hiver qu'en été.

On s'est servi dans l'étude de la méthode GAM afin d'expliquer le multiplicateur par plus ou moins de variables aléatoires du problème. Ainsi, dans l'expérience 2, on explique le prix $\boldsymbol{\lambda}_t$ par le temps t, la demande $\boldsymbol{D}_t$, et la disponibilité thermique $\overline{\boldsymbol{P}}_t$. On remplace ainsi, lors de la résolution des sous-problèmes, la variable $\boldsymbol{\lambda}_t$ par une quantité s'écrivant $f_2^1(t) + f_2^2(\boldsymbol{D}_t) + f_2^3(\overline{\boldsymbol{P}}_t)$. Les fonctions f_2^1, f_2^2 et f_2^3 sont estimées par la méthode GAM et représentées dans la figure 4.10. On observe clairement que le coût marginal est une fonction croissante de la demande, ce qui est tout à fait logique : quand la demande est forte, on se retrouve à devoir utiliser les moyens de production les plus onéreux. Ensuite, plus la disponibilité thermique est grande, moins on risque d'avoir besoin de moyens onéreux, plus le coût marginal du système est faible. On garde, finalement, une influence non négligeable du temps sur le coût marginal.

Enfin, afin de mieux situer les résultats obtenus par l'algorithme DADP, on compare dans la figure 4.11 la fonction de répartition des coûts obtenus en simulation à ceux associés à la borne inférieure, à la borne supérieure, et à la méthode de décomposition par agrégation présentée au §4.3.2. Il apparaît clairement, sur ce graphique, que les ex-

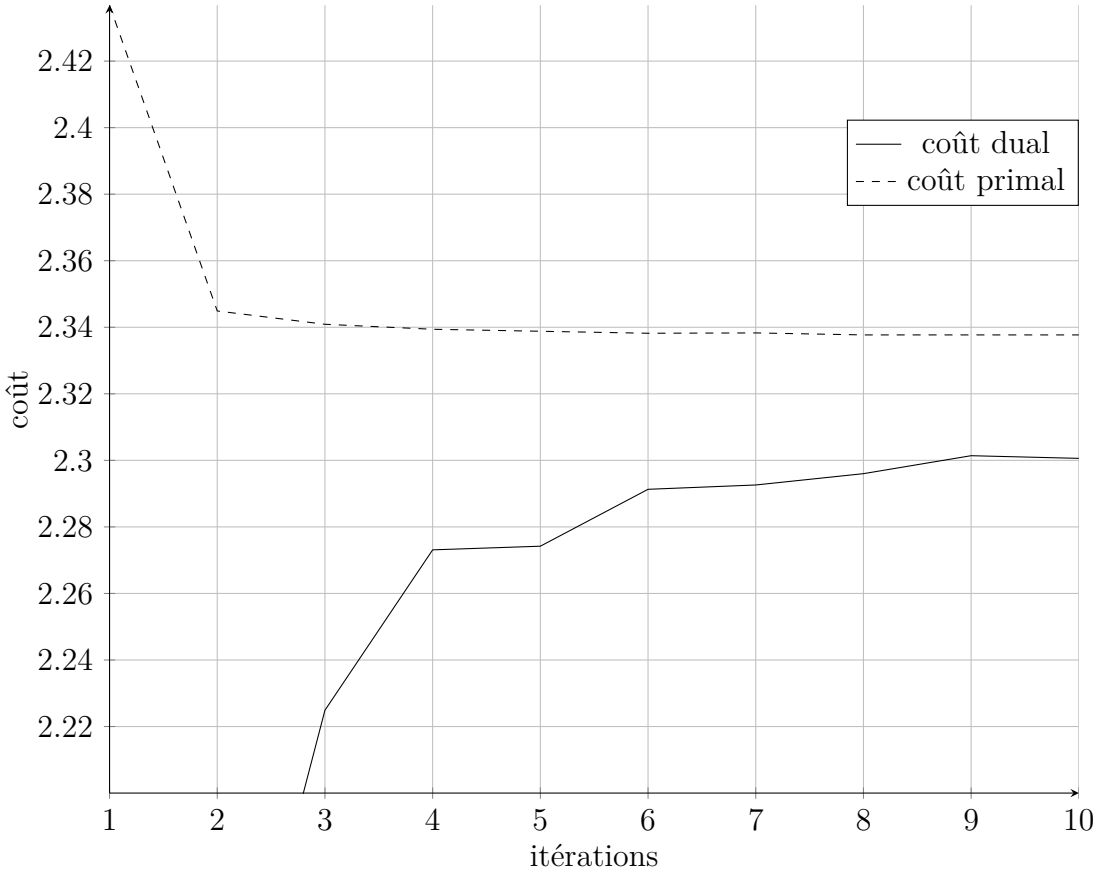

FIGURE 4.6 – Évolution des coûts primal et dual au cours des itérations pour l'expérience 2

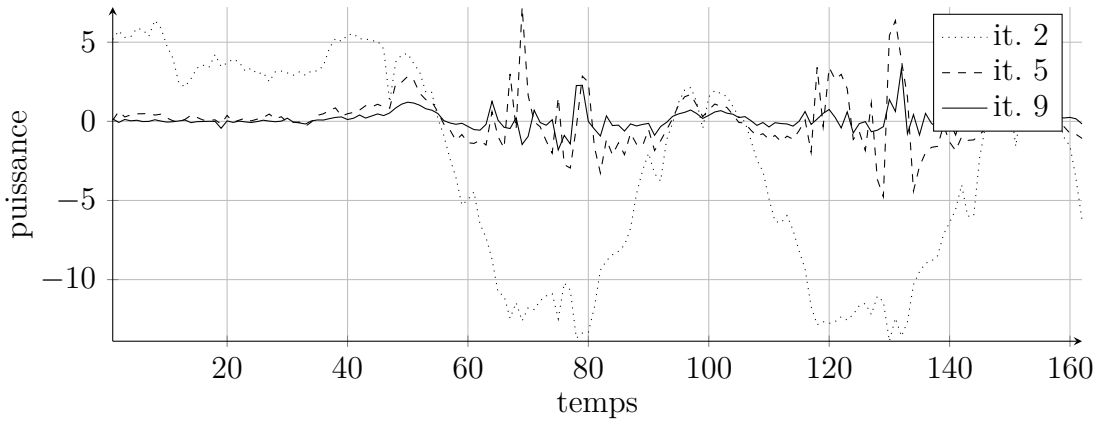

FIGURE 4.7 – Espérance de l'écart production-demande pour l'expérience 2

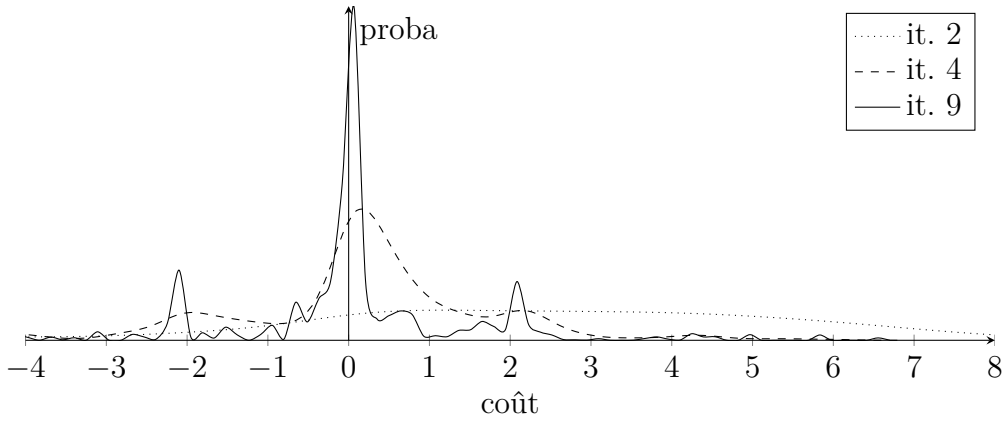

FIGURE 4.8 – Distribution de l'écart production-demande à un pas de temps particulier pour l'expérience 2

périences 1 et 2 obtiennent de meilleurs coûts que ceux de l'expérience 0. Cependant, la méthode de décomposition par agrégation obtient une stratégie encore sensiblement meilleure. Nous pouvons citer deux raisons principales à cela :

1. la méthode par agrégation calcule des fonctions de Bellman de dimension 2 (stock et stock complémentaire) alors que nous n'avons testé, avec DADP, que l'usage de fonctions de Bellman en dimension 1 ;

2. nous n'avons pas essayé de manière exhaustive toutes les possibilités pour la variable d'explication du prix dans l'algorithme DADP. Peut-être existe-t-il une statistique permettant d'obtenir un coût similaire, voire meilleur, que la méthode par agrégation. Par exemple, la dynamique exacte du multiplicateur exhibée sur un exemple par Strugarek (2006) imposerait à la variable d'information que nous introduisons ici de retenir, à chaque instant t, la demande à l'instant précédent $\boldsymbol{D}_{t-1}$. Nous aurions alors à résoudre les sous-problèmes par programmation dynamique en dimension 2. Cela reste à tester.

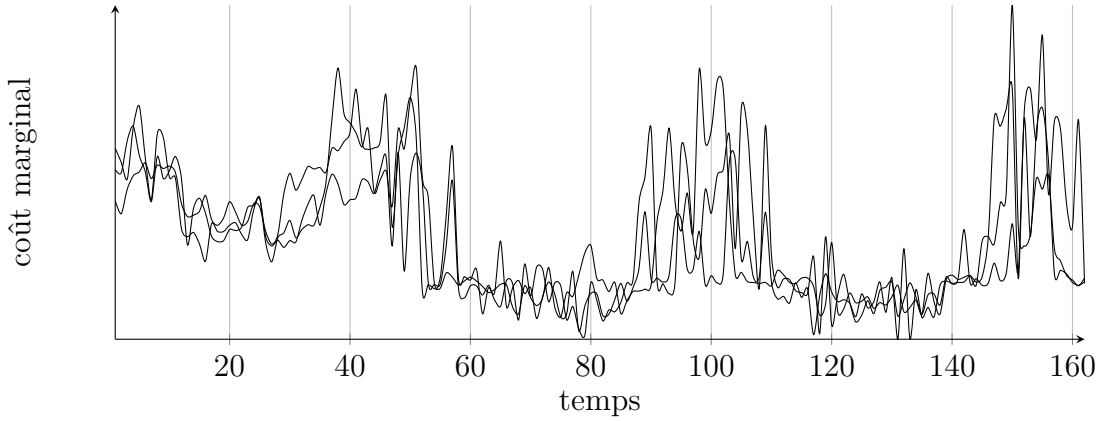

FIGURE 4.9 – Quelques scénarios de coûts marginaux à l'issue de l'expérience 2

Conclusion

Nous présentons un problème de gestion de stocks pour la production électrique. La difficulté principale de ce problème, dans le contexte de la programmation dynamique, est le nombre de réserves, ici 7, qui empêche la résolution numérique directe de l'équation de programmation dynamique. Nous présentons le résultat de la mise en œuvre d'une méthode alternative : l'algorithme *Dual Approximate Dynamic Programming* (DADP) introduit au chapitre 3. On observe le bon comportement de la méthode : le coût associé aux stratégies sorties par l'algorithme diminue lorsque l'on raffine notre approximation, et l'estimation de la variable de coordination s'améliore.

Pour les expériences que nous avons menées, nous obtenons des résultats du même ordre de grandeur, bien que légèrement inférieurs à la méthode de décomposition par agrégation. Ces résultats sont cependant encourageants pour plusieurs raisons.

- Nous n'avons testé que l'usage de fonctions de Bellman de dimension 1 (ne dépendant que du stock local) alors que la méthode de décomposition par agrégation utilise des fonctions de Bellman de dimension 2 (dépendant du stock et du stock complémentaire). Cela nous permet en particulier d'avoir des temps de calcul sensiblement inférieurs à la méthode d'agrégation.
- Nous n'avons pas de procédure automatique de recherche d'une "bonne variable" explicative du prix, et nous avons choisi la demande et l'indisponibilité thermique parmi un grand nombre d'autres choix possibles. Il est donc fort probable que l'on puisse construire une autre variable qui permette d'obtenir de meilleures stratégies.
- Enfin, la méthode par agrégation est très liée à la structure du problème considéré, en particulier au fait que les stocks sont relativement homogènes. Or, les évolutions récentes du marché de l'énergie, notamment l'intégration de contraintes de stock de polluants, nous oblige à considérer de plus en plus des stocks de nature hétérogène. Dans ce contexte, un algorithme primal-dual tel que DADP semble plus adapté.

Ainsi, il serait bon de mener plus loin les expériences numériques en testant d'autres variables explicatives pour le prix que celles utilisées ici. Il serait en particulier souhaitable d'ajouter de la mémoire sur la variable explicative comme cela a été présenté au §3.2.1.

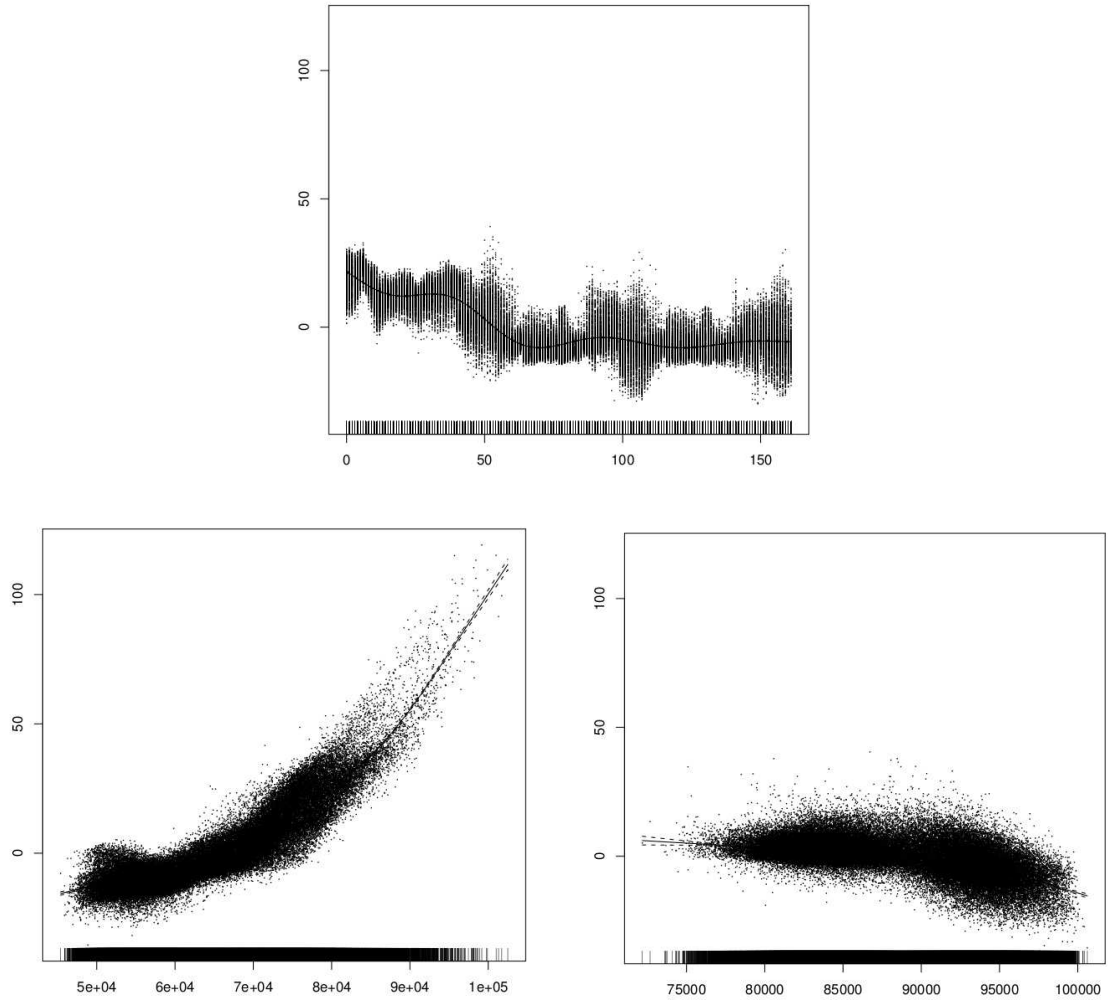

FIGURE 4.10 – Expérience 2 : espérance du prix (en ordonnée) conditionnellement au temps (en haut), à la demande (en bas à gauche) et à la disponibilité thermique (en bas à droite)

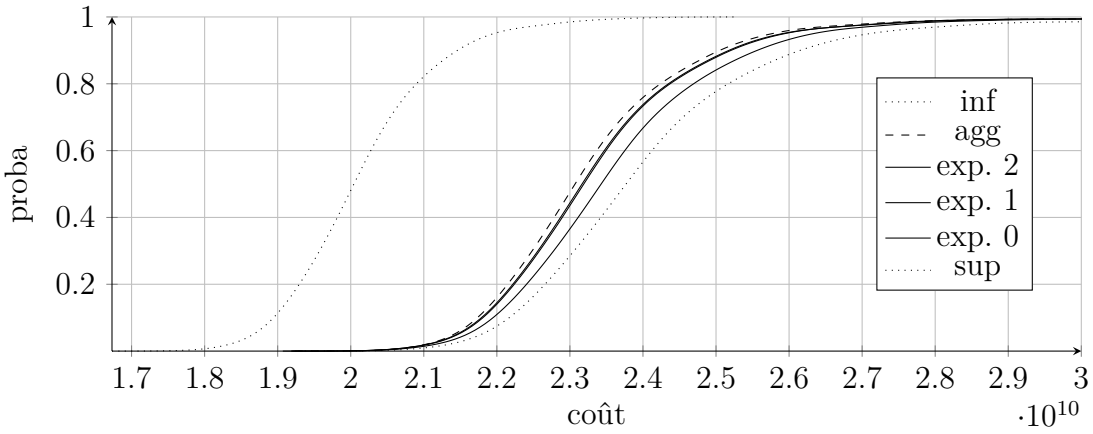

FIGURE 4.11 – Fonctions de répartition des coûts obtenus en simulation par les différentes méthodes

Chapitre 5

Consistance dynamique pour les problèmes de commande optimale stochastique

> However, the thought was finally forced upon me that the desired solution in a control process was a policy: 'Do thus-and-thus if you find yourself in this portion of state space with this amount of time left.' Conversely, once it was realized that the concept of policy was fundamental in control theory, the maticization of the basic engineering concept of 'feedback control', then the emphasis upon a state variable formulation became natural.
>
> Richard Bellman (1920-1984)
> *extrait par Dreyfus (2002) de son autobiographie*

Ce chapitre reprend les idées développées dans un travail commun (Carpentier, Chancelier, Cohen, De Lara, et Girardeau, 2010) publié sur le site *Optimization Online* et soumis à *Annals of Operations Research* en mai 2010.

Pour une suite de problèmes d'optimisation, on introduit de manière informelle la notion de consistance dynamique comme suit. On se pose un premier problème d'optimisation sur un horizon discret et fini, que l'on résout. On obtient ainsi des règles de décision optimales pour le premier instant, disons t_0, ainsi que pour le restant de l'horizon, disons $t_1, \ldots, t_N = T$. On applique alors la décision optimale à l'instant t_0. À l'instant suivant t_1, on est en général capable de poser un nouveau problème d'optimisation, de même nature que le précédent, mais portant sur l'horizon restant $t_1, \ldots, T$. On obtient en résolvant ce nouveau problème de nouvelles règles de décision optimales pour l'instant t_1 et pour le restant de l'horizon. On répète ainsi le processus jusqu'à la fin de l'horizon. On dit que la suite de problèmes d'optimisation ainsi posés est consistante dynamiquement si les règles de décision obtenues à l'issue de la résolution du premier problème restent

optimales pour les problèmes suivants. Autrement dit, les règles de décision qui sont optimales pour le problème au premier instant n'ont pas à être remises en cause aux instants ultérieurs.

Remarque 5.1 (Horizon glissant). On se place dans un cadre où l'horizon de temps est supposé lointain, mais fixe : lors de la formulation du problème d'optimisation à un instant intermédiaire, on considère le même instant final T que dans le problème de départ. On pourrait également se poser la question de la consistance dynamique pour un modèle de décision à horizon glissant, c'est-à-dire où l'horizon recule au fur et à mesure que l'on avance dans le temps. Cette question est sensiblement différente de celle que nous adressons ici.

L'étude de telles propriétés pour des problèmes économiques est assez ancienne, et elle connait un regain important depuis quelques années au sein de la communauté *Stochastic Programming*, où l'on s'intéresse au remplacement, dans le critère d'optimisation, de la "classique" espérance par une mesure de risque plus générale. Nous proposons ici de dresser un parallèle qui nous paraît essentiel entre la notion de consistance dynamique et le concept de variable d'état dans le cadre du Principe de programmation dynamique.

5.1 État de l'art

La notion de consistance dynamique telle que nous venons de la présenter informellement est bien connue des économistes (voir Strotz, 1955, Kreps et Porteus, 1978, Hammond, 1989). Elle se pose ainsi généralement à une suite de problèmes de décision. Le même type de question se pose lorsque l'on s'intéresse à quantifier les risques associés à des revenus incertains. C'est l'objet de l'étude des mesures de risques. La littérature autour des mesures de risque s'est d'abord intéressée à définir les propriétés souhaitables d'une mesure afin qu'elle soit sensée du point de vue économique. Considérons un espace de probabilité $(\Omega, \mathcal{A}, \mathbb{P})$. On appelle *mesure de risque* une application ρ définie sur un sous-ensemble de variables aléatoires réelles, que l'on peut voir comme des revenus incertains, à valeurs dans $\overline{\mathbb{R}}$. La notion de mesure de risque *cohérente*[1] a été introduite par Artzner, Delbaen, Eber, et Heath (1999) pour décrire les propriétés qu'il convient de demander à une mesure de risque "raisonnable" :

$$\begin{aligned}
&\text{invariance par translation :} && \rho(\boldsymbol{Z} + a) = \rho(\boldsymbol{Z}) + a,\\
&\text{sous-additivité :} && \rho(\boldsymbol{Z} + \boldsymbol{Z}') \leq \rho(\boldsymbol{Z}) + \rho(\boldsymbol{Z}'),\\
&\text{homogénéité positive :} && \rho(\lambda \boldsymbol{Z}) = \lambda \rho(\boldsymbol{Z}), \qquad \forall \lambda \geq 0,\\
&\text{monotonie :} && \boldsymbol{Z} \leq \boldsymbol{Z}' \Rightarrow \rho(\boldsymbol{Z}) \geq \rho(\boldsymbol{Z}').
\end{aligned}$$

Ainsi, par exemple, la propriété de monotonie indique que si un portefeuille $\boldsymbol{Z}$ a toujours des revenus inférieurs à ceux d'un portefeuille $\boldsymbol{Z}'$, alors le risque associé à $\boldsymbol{Z}$ est toujours plus grand que celui associé à $\boldsymbol{Z}'$. La sous-additivité traduit quant à elle la propriété liée à la diversification : le risque associé à un portefeuille diversifié (constitué de plusieurs portefeuilles) ne peut pas être plus important que la somme des risques des portefeuilles pris

1. Nous utilisons le terme *cohérence* pour désigner le terme anglais *coherency* et le terme *consistance* pour désigner le terme anglais *consistency*.

séparément. Toutes ces propriétés ont été traduites dans le cadre dynamique de diverses manières par Artzner, Delbaen, Eber, Heath, et Ku (2007), Riedel (2004), Detlefsen et Scandolo (2005), Cheridito, Delbaen, et Kupper (2006).

Au sein de la communauté *Stochastic Programming*, les propriétés liées à la cohérence, tant statique que dynamique, ont été étudiées récemment par Shapiro (2009) et par Ruszczyński (2009) pour les chaînes de Markov contrôlées. Ruszczyński (2009) introduit alors la propriété de consistance dynamique qui permet de lier les problèmes de décision successifs entre eux de la manière décrite intuitivement en introduction de ce chapitre. L'objectif est de remplacer le critère en espérance par un critère faisant intervenir une mesure de risque, et de rechercher les propriétés que l'on doit demander à la mesure de risque pour pouvoir conserver certaines propriétés structurelles du problème d'optimisation, telles que la consistance dynamique. Autrement dit, l'auteur considère un problème de commande optimale stochastique avec critère en espérance qui est naturellement consistant dynamiquement. Puis, il cherche à remplacer le critère en espérance par un critère faisant intervenir des mesures de risque en choisissant celles-ci de telle sorte que la suite de problèmes de décision reste consistante dynamiquement.

Le point de vue que nous adoptons ici est différent. Nous considérons des suites de problèmes de décision, faisant intervenir des contraintes de risque et nous montrons que, quitte à changer d'état pour le système, la suite de problèmes de décision reste consistante dynamiquement. En d'autres termes, nous ne restreignons pas le type de contraintes de risque à considérer, et nous montrons qu'il est possible de préserver la notion de consistance dynamique.

5.2 Parallèle avec la programmation dynamique

Nous mettons en lumière dans cette section les liens étroits qui existent entre la notion de consistance dynamique et le concept de variable d'état pour un système dynamique. Au §5.2.1, nous faisons un certain nombre de constats à ce sujet pour un exemple déterministe, constats que l'on retrouve ensuite aux §5.2.2 et §5.2.3 dans le cas stochastique.

5.2.1 Un exemple déterministe

Nous nous concentrons ici sur la notion de consistance dynamique pour une suite de problèmes de commande optimale déterministe. La discussion reste relativement informelle au sens où nous ne prétendons pas fournir toutes les hypothèses techniques garantissant par exemple l'existence d'une solution pour les problèmes que nous considérons. Le but de cette section est de présenter dans un cadre relativement simple des propriétés structurelles importantes qui resteront vraies dans un cadre plus complexe, où elles seraient plus difficiles à appréhender directement.

On considère un intervalle de temps discret et fini $t_0, \dots, t_N = T$ [2]. L'objectif est d'optimiser sur cet horizon la gestion d'un stock qui, à l'instant t, prend la valeur x_t appartenant à un certain espace $\mathcal{X}_t$ (par exemple $\mathbb{R}^n$), à l'aide de décisions $u_t \in \mathcal{U}_t$ (par exemple $\mathbb{R}^m$) influant sur le système. À chaque instant, un coût $L_t(x_t, u_t)$ est infligé au système et le stock évolue à l'aide de la dynamique $x_{t+1} = f_t(x_t, u_t)$. Le but du décideur est de minimiser la somme des coûts intermédiaires L_t et d'un coût final K, ce qui s'écrit :

2. On adoptera la convention : $t_{i+1} = t_i + 1$.

$$\min_{x,u} \quad \sum_{t=t_0}^{T-1} L_t\left(x_t, u_t\right) + K\left(x_T\right), \tag{5.1a}$$

sous les contraintes de dynamique :

$$x_{t_0} \text{ donné}, \tag{5.1b}$$
$$x_{t+1} = f_t\left(x_t, u_t\right), \qquad \forall t = t_0, \ldots, T-1. \tag{5.1c}$$

Une solution possible à ce problème est une suite (en temps) de décisions à appliquer. Ce processus de décision est généralement qualifié de boucle ouverte. Mais cette suite de décisions dépend bien sûr de la condition initiale x_{t_0} du problème. On peut ainsi dire que la décision à l'instant t est prise en *feedback* sur le temps et sur la condition initiale x_{t_0}.

Notons $u^\sharp_{t_0,t_0}, \ldots, u^\sharp_{t_0,T-1}$ une solution au problème (5.1). Le premier indice nous indique que le problème dont ces commandes sont solutions est posé partant de t_0 ; le deuxième indice repère l'instant auquel la décision est prise. Supposons qu'il existe également une solution pour chacun des problèmes se posant naturellement aux pas de temps suivants $t_i = t_1, \ldots, T-1$:

$$\min_{x,u} \quad \sum_{t=t_i}^{T-1} L_t\left(x_t, u_t\right) + K\left(x_T\right), \tag{5.2a}$$

sous les contraintes de dynamique :

$$x_{t_i} \text{ donné}, \tag{5.2b}$$
$$x_{t+1} = f_t\left(x_t, u_t\right), \qquad \forall t = t_i, \ldots, T-1. \tag{5.2c}$$

Nous notons ces solutions $u^\sharp_{t_i,t_i}, \ldots, u^\sharp_{t_i,T-1}$. De la même manière, le premier indice nous indique que le problème dont ces commandes sont solutions est posé partant de t_i ; le deuxième indice repère l'instant auquel la décision est prise. Il faut garder à l'esprit que les notations utilisées ici masquent le fait que ces solutions dépendent généralement de la condition initiale x_{t_i}. Ceci nous inspire une première observation.

Lemme 5.1 (Indépendance de la condition initiale)**.** *Dans le cas très particulier où les solutions des problèmes* (5.1) *et* (5.2) *pour* $t_i = t_1, \ldots, T-1$ *sont indépendantes des conditions initiales, la suite de problèmes est consistante dynamiquement.*

Démonstration. Le principe d'optimalité de Bellman appliqué au problème (5.1) nous indique que la solution du problème (5.2) avec $x_{t_i} = x^\sharp_{t_0,t_i}$ est $u^\sharp_{t_0,t_i}, \ldots, u^\sharp_{t_0,T-1}$. Par hypothèse, c'est aussi la solution pour toute valeur de x_{t_i}. La suite de décisions optimales issue du problème au premier instant (5.1) est donc optimale pour tous les problèmes d'optimisation suivants. □

Cette propriété d'indépendance de la condition initiale est bien sûre fausse en général. Nous observerons cependant au §5.2.2 que nombre de problèmes rencontrés en pratique satisfont cette propriété surprenante.

Exemple 5.1. Soit, pour tout $t = t_0, \dots, T-1$, les fonctions $l_t : \mathcal{U}_t \to \mathbb{R}$ et $f_t : \mathcal{U}_t \to \mathbb{R}$, et K un scalaire. On supposera de plus le stock x_t scalaire. On considère le problème de contrôle optimal déterministe suivant :

$$\begin{aligned} \min_{x,u} \quad & \sum_{t=t_0}^{T-1} l_t(u_t)\, x_t + K x_T, \\ \text{s.c.} \quad & x_{t_0} \text{ donné}, \\ & x_{t+1} = f_t(u_t)\, x_t, \qquad \forall t = t_0, \dots, T-1. \end{aligned}$$

Remarquons que les variables x_t peuvent être remplacées de manière récursive à l'aide des dynamiques f_t, de sorte que le problème ci-dessus peut être écrit de manière équivalente :

$$\min_{u} \quad \sum_{t=t_0}^{T-1} l_t(u_t)\, f_{t-1}(u_{t-1}) \dots f_{t_0}(u_{t_0})\, x_{t_0} + K f_{T-1}(u_{T-1}) \dots f_{t_0}(u_{t_0})\, x_{t_0}.$$

Ainsi le coût optimal du problème est-il linéaire par rapport à la condition initiale x_{t_0}. Supposons que x_{t_0} ne prenne que des valeurs positives. Alors sa valeur n'a aucune influence sur l'arg min de ce critère (il influence seulement la valeur du problème). Formulons alors le même problème à une date ultérieure $t_i = t_1, \dots, T-1$, avec comme condition initiale x_{t_i}. Par le même argument que pour le problème au premier pas de temps, alors la valeur de x_{t_i} n'a pas d'influence sur le contrôle optimal, si on suppose que les dynamiques sont telles que le stock x_t reste positif pour tout $t = t_1, \dots, T$. Les hypothèses du Lemme 5.1 sont donc satisfaites, et la propriété de consistance dynamique est donc vraie.

Comme nous l'avons fait remarquer, le Lemme 5.1 ne s'applique pas en général. De plus, la formulation déterministe (5.1) provient souvent de la représentation d'un processus de décision qui est en fait soumis à des perturbations non modélisées. Nous avons à l'esprit un contexte industriel dans lequel des décisions sont prises de façon séquentielle de la manière suivante.

- À l'instant t_0, le problème (5.1) est résolu. On obtient ainsi une décision $u^\sharp_{t_0,t_0}$ qui s'applique à l'instant t_0, et une suite de décisions $u^\sharp_{t_0,t_1}, \dots, u^\sharp_{t_0,T-1}$ qui concernent les pas de temps ultérieurs.
- À l'instant t_1, on observe le stock et on écrit le problème commençant en t_1 avec comme condition initiale $x_{t_1} = f_{t_0}(x_{t_0}, u^\sharp_{t_0,t_0}) + \varepsilon_{t_1}$, où ε_{t_1} est une perturbation que le modèle initial n'avait pas pris en compte. Il n'y a pas de raison de ne pas utiliser cette observation du stock plutôt que la valeur du stock qui était attendue.
- Ainsi est obtenue une décision $u^\sharp_{t_1,t_1}$, généralement différente de celle calculée initialement, qui était $u^\sharp_{t_0,t_1}$.
- Le même processus se poursuit jusqu'à la fin de l'horizon de temps.

Nous pouvons maintenant énoncer les deux lemmes suivants.

Lemme 5.2 (Monde parfaitement déterministe)**.** *Dans le cas où le modèle est exact, c'est-à-dire si les perturbations ε_{t_i} introduites plus haut sont identiquement nulles, les problèmes* (5.2) *avec comme conditions initiales $x_{t_i} = x^\sharp_{t_i} := f_{t_i}(x^\sharp_{t_{i-1}}, u^\sharp_{t_0,t_{i-1}})$ sont consistants dynamiquement.*

Démonstration. Il s'agit d'une application immédiate du principe d'optimalité de Bellman au problème (5.1). □

Il est bien évident que le Lemme 5.2 n'est pas vérifié dans le monde réel qui, comme chacun sait, n'est pas déterministe. Ainsi, il semble que les perturbations (imprévues) qui affectent le système apportent de l'inconsistance à la suite de problèmes de décisions : les décisions optimales vues du premier pas de temps ne le sont plus lorsque l'on pose à nouveau le problème à un instant ultérieur.

En fait, comme nous le faisons maintenant remarquer, la consistance dynamique est préservée si on prend soin de considérer des stratégies dépendant d'une information suffisante.

Lemme 5.3 (Quantité suffisante d'information)**.** *Supposons que nous résolvions le problème* (5.1) *par programmation dynamique, obtenant ainsi une suite de stratégies optimales* $\Phi^\sharp_{t_0,t_0}, \ldots, \Phi^\sharp_{t_0,T-1}$ *en feedback sur la variable* x*. Alors les problèmes* (5.2) *sont consistants dynamiquement pour* $t = t_0, \ldots, T-1$*.*

Démonstration. Le résultat est une application directe du principe de programmation dynamique, qui établit précisément qu'il existe une telle fonction $\Phi^\sharp_{t_0,t_i}$ qui est optimale à la fois pour le problème (5.1) et pour les problèmes (5.2) aux différents instants t_i, quelles que soient les conditions initiales x_{t_i}. □

On récupère donc la propriété de consistance dynamique dès lors que sont utilisées les stratégies $\Phi^\sharp_{t_0,t}$ plutôt que de simples contrôles $u^\sharp_{t_0,t}$. En d'autres termes, les problèmes sont consistants dynamiquement si on fait en sorte que la décision soit basée sur une information suffisante (ici l'instant t *et* la variable x).

Il y a bien sûr un lien entre les stratégies optimales et les contrôles $(u^\sharp_{t_0,t_0}, \ldots, u^\sharp_{t_0,T-1})$:

$$u^\sharp_{t_0,t} = \Phi^\sharp_{t_0,t}\left(x^\sharp_{t_0,t}\right), \qquad \forall t = t_0, \ldots, T-1,$$

avec

$$\begin{aligned} x^\sharp_{t_0,t_0} &= x_{t_0}, \\ x^\sharp_{t_0,t+1} &= f_t\left(x^\sharp_{t_0,t}, \Phi^\sharp_{t_0,t}\left(x^\sharp_{t_0,t}\right)\right), \qquad \forall t = t_0, \ldots, T-1. \end{aligned}$$

Le constat que nous faisons ici, et qui peut certes sembler trivial pour le moment, se résume ainsi : dans nombre de cas, la suite de problèmes d'optimisation est dynamiquement consistante pourvu qu'on base les décisions sur une information suffisante.

5.2.2 Commande optimale stochastique sans contrainte de risque

Considérons maintenant le cas plus général où le système dynamique est soumis à des perturbations exogènes prises en compte dans le modèle. Le décideur cherche des stratégies permettant de gérer le système de façon à minimiser une certaine fonction objectif sur le même horizon de temps que précédemment. Il s'agit d'un problème d'optimisation dynamique sur lequel on peut se poser la question de la consistance temporelle. Comme dans le problème précédent, la famille de problèmes se déduit de celui au premier pas de temps en tronquant la fonction coût et les dynamiques à partir du pas de temps courant. De plus, les stratégies sont toutes définies sur la même structure d'information que dans le premier problème.

On considère un système dynamique portant sur un processus de stock $\boldsymbol{X}$, prenant la valeur $\boldsymbol{X}_t$ à l'instant t, qui est maintenant un processus stochastique. Ce stock prend à l'instant t ses valeurs dans un certain espace $\mathcal{X}_t$. Le système est influencé par des variables de contrôle $\boldsymbol{U} := (\boldsymbol{U}_t)_{t=t_0,\dots,T-1}$ ainsi que par des bruits exogènes au système $\boldsymbol{W} := (\boldsymbol{W}_t)_{t=t_0,\dots,T}$ ($\boldsymbol{U}_t$ et $\boldsymbol{W}_t$ prennent leurs valeurs dans des espaces $\mathcal{U}_t$ et $\mathcal{W}_t$, respectivement).

Toutes les variables aléatoires sont définies sur un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$. Le problème que nous considérons consiste à minimiser l'espérance de la somme de coûts dépendant du stock, du contrôle et du bruit sur un horizon de temps discret et fini. L'évolution de la variable de stock est régie par une dynamique qui, à chaque pas de temps, dépend des valeurs courantes du stock, du contrôle et du bruit. Le problème commençant à l'instant t_0 s'écrit :

$$\min_{\boldsymbol{X},\boldsymbol{U}} \quad \mathbb{E}\left(\sum_{t=t_0}^{T-1} L_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right) + K\left(\boldsymbol{X}_T\right)\right), \tag{5.3a}$$

sous les contraintes de dynamique :

$$\boldsymbol{X}_{t_0} \text{ donné}, \tag{5.3b}$$
$$\boldsymbol{X}_{t+1} = f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right), \qquad \forall t = t_0, \dots, T-1, \tag{5.3c}$$

et la contrainte de non-anticipativité :

$$\boldsymbol{U}_t \preceq \boldsymbol{X}_{t_0}, \boldsymbol{W}_{t_1}, \dots, \boldsymbol{W}_t, \qquad \forall t = t_0, \dots, T-1. \tag{5.3d}$$

Les bruits affectant le système peuvent être corrélés en temps. Cependant, il est classique en commande optimale stochastique d'avoir recours à un processus dit de blanchiment du bruit, qui revient à inclure dans la variable $\boldsymbol{X}_t$ l'information suffisante pour que les bruits soient indépendants en temps. Par exemple, supposons que le bruit soit un processus réel tel que $\boldsymbol{W}_{t+1} = \alpha \boldsymbol{W}_t + \boldsymbol{\varepsilon}_{t+1}$ avec $\boldsymbol{\varepsilon}_{t+1}$ indépendant de $\boldsymbol{X}_{t_0}, \boldsymbol{W}_{t_1}, \dots, \boldsymbol{W}_t$, et ce pour tout $t = t_1, \dots, T-1$. Dans ce cas, on posera comme "nouvelle variable de stock" : $\boldsymbol{Y}_t = (\boldsymbol{X}_t, \boldsymbol{W}_t)$ et comme "nouveau bruit" $\boldsymbol{\varepsilon}_{t+1}$, de sorte que l'on aura la nouvelle dynamique de stock :

$$\boldsymbol{Y}_{t+1} := \begin{pmatrix} \boldsymbol{X}_{t+1} \\ \\ \boldsymbol{W}_{t+1} \end{pmatrix} = \begin{pmatrix} f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \alpha \boldsymbol{W}_t + \boldsymbol{\varepsilon}_{t+1}\right) \\ \\ \alpha \boldsymbol{W}_t + \boldsymbol{\varepsilon}_{t+1} \end{pmatrix}.$$

Les variables aléatoires $\boldsymbol{X}_{t_0}, \boldsymbol{\varepsilon}_{t_1}, \dots, \boldsymbol{\varepsilon}_T$ qui constituent le "nouveau" processus de bruit sont maintenant indépendantes en temps. Cette procédure de blanchiment fait qu'au plus, c'est-à-dire si $\boldsymbol{W}_{t+1}$ dépend de tout la passé du bruit, on devra inclure tout le passé du bruit dans la variable $\boldsymbol{X}$. On remarque que de plus que cette procédure conserve, pour notre problème, la propriété d'information parfaite.

Ainsi, quitte à avoir préalablement blanchi le processus de bruit, nous faisons l'hypothèse suivante.

Hypothèse 5.4 (Cas markovien)**.** Les variables aléatoires $\boldsymbol{X}_{t_0}, \boldsymbol{W}_{t_1}, \dots, \boldsymbol{W}_T$ sont indépendantes entre elles.

Sous l'hypothèse 5.4, il est bien connu (voir Bertsekas, 2000) que :

– il n'y pas de perte d'optimalité à rechercher la commande $\boldsymbol{U}_t$ à l'instant t sous la forme d'une fonction en *feedback* sur la variable $\boldsymbol{X}_t$, c'est-à-dire une fonction de la forme $\Phi_{t_0,t} : \mathcal{X}_t \to \mathcal{U}_t$;
– les stratégies optimales $\Phi^\sharp_{t_0,t_0}, \ldots, \Phi^\sharp_{t_0,T-1}$ peuvent être obtenues par résolution de l'équation de programmation dynamique. Soit $V_t(x)$ le coût optimal partant de l'instant t au stock x, l'équation en question s'écrit :

$$\begin{aligned} V_T(x) &= K(x), \\ V_t(x) &= \min_{u \in \mathcal{U}_t} \mathbb{E}\Big(L_t(x, u, \boldsymbol{W}_{t+1}) + V_{t+1}\big(f_t(x, u, \boldsymbol{W}_{t+1})\big)\Big). \end{aligned}$$

Nous sommes dans le cas le plus standard en commande optimale stochastique. Par construction, il est clair que les stratégies optimales $\Phi^\sharp_{t_0,t_0}, \ldots, \Phi^\sharp_{t_0,T-1}$ issues de la résolution de l'équation de programmation dynamique restent optimales pour les problèmes aux instants ultérieurs $t_i = t_1, \ldots, T-1$:

$$\min_{\boldsymbol{X},\boldsymbol{U}} \quad \mathbb{E}\left(\sum_{t=t_i}^{T-1} L_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right) + K\left(\boldsymbol{X}_T\right) \right), \tag{5.4a}$$

sous les contraintes de dynamique :

$$\boldsymbol{X}_{t_i} \text{ donné}, \tag{5.4b}$$
$$\boldsymbol{X}_{t+1} = f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right), \qquad \forall t = t_i, \ldots, T-1, \tag{5.4c}$$

et la contrainte de non-anticipativité :

$$\boldsymbol{U}_t \preceq \boldsymbol{X}_{t_i}, \boldsymbol{W}_{t_{i+1}}, \ldots, \boldsymbol{W}_t, \qquad \forall t = t_i, \ldots, T-1. \tag{5.4d}$$

En d'autres termes, ces problèmes sont consistants dynamiquement dès lors que la variable sur laquelle la décision se base à l'instant t contient $\boldsymbol{X}_t$. On peut alors faire l'analogie avec le cas déterministe précédemment étudié. Cependant, le lecteur doit être conscient que le cas présent est plus proche du cadre du lemme 5.1 que de celui du lemme 5.3. Nous en expliquons à présent la raison.

Pour ce faire, nous présentons une formulation équivalente du problème stochastique qui fait le lien avec le problème déterministe précédent. On trouvera des détails sur cette formulation dans les travaux de Witsenhausen (1973).

Soit Ψ_t l'ensemble des fonctions de $\mathcal{X}_t$ à valeurs dans $\mathbb{R}$. On note μ_{t_0} la loi de l'état initial $\boldsymbol{X}_{t_0}$ et on se donne une suite de *feedbacks* $\Phi_t : \mathcal{X}_t \to \mathcal{U}_t$, pour tout $t = t_0, \ldots, T-1$. On peut alors définir l'opérateur $A_t^{\Phi_t} : \Psi_{t+1} \to \Psi_t$ de la manière suivante : [3]

$$\left(A_t^{\Phi_t}\psi_{t+1}\right)(\cdot) := \mathbb{E}\Big(\psi_{t+1} \circ f_t\left(\cdot, \Phi_t\left(\cdot\right), \boldsymbol{W}_{t+1}\right)\Big),$$

où l'espérance dans le membre de droite porte sur la variable aléatoire $\boldsymbol{W}_{t+1}$. Étant donné un *feedback* Φ_t et une fonction de coût $\psi_{t+1} \in \Psi_{t+1}$, pour tout $x \in \mathcal{X}_t$ la valeur $(A_t^{\Phi_t}\psi_{t+1})(x)$ est l'espérance conditionnelle de la quantité $\psi_{t+1}(\boldsymbol{X}_{t+1})$, sachant $\boldsymbol{X}_t = x$ pour le *feedback* Φ_t.

3. Nous ne discutons pas ici des hypothèses techniques nécessaires à l'intégrabilité notamment. Nous supposons que les quantités que nous définissons existent.

On définit, pour tout $\psi_t \in \Psi_t$ et toute loi de probabilité μ_t sur $\mathcal{X}_t$, la quantité $\langle \psi_t, \mu_t \rangle$ comme $\mathbb{E}\left(\psi_t(\boldsymbol{X}_t)\right)$ lorsque $\boldsymbol{X}_t$ est distribué selon la loi μ_t. Remarquons que l'opérateur $A_t^{\Phi_t}$ est linéaire. On définit son opérateur adjoint $(A_t^{\Phi_t})^\star$, c'est-à-dire l'opérateur de l'espace des mesures sur $\mathcal{X}_{t+1}$ vers l'espace des mesures sur $\mathcal{X}_t$ tel que :

$$\left\langle A_t^{\Phi_t} \psi_{t+1}, \mu_t \right\rangle = \left\langle \psi_{t+1}, \left(A_t^{\Phi_t}\right)^\star \mu_t \right\rangle,$$

pour toute mesure μ_t sur l'espace $\mathcal{X}_t$. Notons que l'on peut alors introduire, via un argument de dualité, l'équation de Fokker-Planck régissant l'évolution de la loi de probabilité de l'état (contrôlée par les lois de *feedback* Φ_t) :

$$\mu_{t+1} = \left(A_t^{\Phi_t}\right)^\star \mu_t,$$

avec $(A_t^{\Phi_t})^\star$ l'adjoint de l'opérateur $A_t^{\Phi_t}$. Nous avons également besoin d'introduire un opérateur $\Lambda_t^{\Phi_t} : \mathcal{X}_t \to \mathbb{R}$ de la forme suivante.

$$\Lambda_t^{\Phi_t}(\cdot) := \mathbb{E}\left(L_t\left(\cdot, \Phi_t(\cdot), \boldsymbol{W}_{t+1}\right)\right),$$

qui calcule le coût instantané espéré à l'instant t pour chaque valeur possible de x lorsque le *feedback* Φ_t est appliqué.

Nous pouvons maintenant écrire la formulation distribuée du problème (5.3), qui est un problème de contrôle optimal déterministe en dimension infinie :

$$\min_{\Phi, \mu} \quad \sum_{t=t_0}^{T-1} \left\langle \Lambda_t^{\Phi_t}, \mu_t \right\rangle + \langle K, \mu_T \rangle,$$

sous les contraintes de dynamique :

$$\begin{aligned} &\mu_{t_0} \text{ donné,} \\ &\mu_{t+1} = \left(A_t^{\Phi_t}\right)^\star \mu_t, \qquad \forall t = t_0, \ldots, T-1. \end{aligned}$$

Il s'agit d'un problème linéaire en les variables μ_t, $t = t_0, \ldots, T$, et non-linéaire en les variables Φ_t, $t = t_0, \ldots, T-1$.

Remarque 5.2. Une formulation alternative de ce problème est la suivante.

$$\begin{aligned} \min_{\Phi, \psi} \quad & \langle \psi_{t_0}, \mu_{t_0} \rangle, \\ \text{s.c.} \quad & \psi_T = K \text{ donné,} \\ & \psi_t = A_t^{\Phi_t} \psi_{t+1} + \Lambda_t^{\Phi_t}, \qquad \forall t = T-1, \ldots, t_0. \end{aligned}$$

Du fait que "l'état" $\psi_t(\cdot)$ suit une dynamique (affine) rétrograde en temps, on pourra qualifier cette écriture de formulation rétrograde. Comme la précédente formulation, il s'agit d'un problème de programmation mathématique en dimension infinie, linéaire en une partie des variables, à savoir ψ_t, $t = t_0, \ldots, T-1$. Les deux formulations sont par ailleurs duales l'une de l'autre. Ainsi les fonctions $\mu(\cdot)_t$ et $\psi(\cdot)_t$ sont les états et/ou état adjoints (selon laquelle des deux formulations est considérée comme le problème primal) de ce problème de contrôle optimal déterministe, duquel les Φ_t sont les contrôles distribués.

Remarquons que les lois de probabilité μ_t, qui sont par définition positives, n'apparaissent que de manière multiplicative dans le problème. Ainsi sommes-nous dans un cas similaire à celui de l'exemple 5.1, avec la différence suivante : du fait que nous avons affaire à des lois de probabilité plutôt qu'à des scalaires, nous avons besoin d'utiliser de manière rétrograde en temps une série de théorèmes d'interversion afin de pouvoir échanger les opérateurs d'espérance et de minimisation en présence. Ceci est nécessaire afin de montrer que la solution du problème ne dépend en fait pas de sa condition initiale μ_{t_0}. En effet, supposons que μ_{T-1} est connu à l'instant $T-1$. Alors, le problème d'optimisation le plus intérieur s'écrit :

$$\begin{aligned} \min_{\Phi_{T-1}} \quad & \left\langle \Lambda_t^{\Phi_{T-1}}, \mu_{T-1} \right\rangle + \langle K, \mu_T \rangle , \\ \text{s.c.} \quad & \mu_T = \left(A_{T-1}^{\Phi_{T-1}} \right)^\star \mu_{T-1}, \end{aligned}$$

ce qui s'écrit de manière équivalente :

$$\min_{\Phi_{T-1}} \quad \left\langle \Lambda_t^{\Phi_{T-1}} + A_{T-1}^{\Phi_{T-1}} K, \mu_{T-1} \right\rangle .$$

L'argument est le suivant. Les opérateurs $\Lambda_t^{\Phi_{T-1}} + A_{T-1}^{\Phi_{T-1}} K$ et μ_{T-1} prennent tous deux leurs valeurs dans $\mathcal{X}_{T-1}$ et l'opération de minimisation peut être effectuée "x par x", de telle sorte que nous nous retrouvons dans un cas similaire à celui de l'exemple 5.1 pour chaque x. Ainsi, l'arg min ne dépend pas de μ_{T-1}. Une preuve rigoureuse nécessiterait un certain nombre d'hypothèses techniques de mesurabilité, que nous ne discutons pas ici, mais que le lecteur pourra trouver dans l'ouvrage de Rockafellar et Wets (1998, Théorème 14.60). Le même argument s'applique aux pas de temps précédents de manière récursive de sorte que, à t_0, la condition initiale μ_{t_0} influence uniquement le coût optimal du problème, mais pas l'arg min (qui est ici le *feedback* $\Phi_{t_0,t}^\sharp$). En d'autres termes, les *feedbacks* optimaux ne dépendent pas des μ_t ; ils ne dépendent donc pas de la condition initiale.

Ainsi, de par le lemme 5.1, les problèmes (5.4) sont naturellement consistants dynamiquement lorsque les commandes sont cherchées en *feedback* sur $\boldsymbol{X}_t$. Il semble donc que la classe de problèmes (assez générale) pouvant être formulés à la manière du problème (5.3) est en fait un cas relativement particulier de problème de commande optimale stochastique. Nous allons voir au §5.2.3 que cette propriété disparaît dès lors que l'on ajoute certains ingrédients au modèle.

5.2.3 Commande optimale stochastique avec contraintes

On peut définir informellement le concept de structure d'état à la manière de Whittle (1982, Section 1.1) : il s'agit d'une variable x prenant la valeur x_t à l'instant t qui est suffisante pour que :

1. la décision optimale à l'instant t dépende au plus de x_t et de t ;
2. la variable x_{t_0} soit observable à $t = t_0$ et que x_{t+1} soit calculable à l'aide de x_t, de la décision à l'instant t et de la nouvelle observation à l'instant $t+1$.

Une définition plus rigoureuse apparait à travers le principe d'optimalité de Bellman tel qu'il est énoncé, par exemple, par Whittle (1982, Section 11.1) et rappelé dans le théorème 1.4.

Nous venons de voir au §5.2.2 que, pour la classe de problèmes que nous avons qualifié de commande optimale stochastique sans contrainte, la variable d'état naturelle était $\boldsymbol{X}_t$. Nous donnons maintenant un exemple de modèle dans lequel la variable d'état ne peut être réduite à $\boldsymbol{X}_t$ comme précédemment. Il s'agit en fait d'ajouter une contrainte au modèle et de constater que la structure d'état change. Dans les problèmes d'optimisation stochastique dynamique, on peut rencontrer différents types de contraintes, telles que[4] les contraintes presque-sûres du type :

$$g\left(\boldsymbol{X}_T\right) \leq a, \tag{5.5}$$

ainsi que des contraintes en probabilité, ou encore des contraintes en espérance dont nous parlons plus bas. Les contraintes presque-sûres sont bien étudiées et on sait qu'elles ne modifient pas la nature du problème d'optimisation. Notant $\mathcal{X}_T^{\text{ad}}$ l'ensemble des variables aléatoires $\boldsymbol{X}_T$ (à valeurs dans $\mathbb{X}_T$) satisfaisant (5.5) et $\mathbb{X}_T^{\text{ad}}$ l'ensemble $\{x \in \mathbb{X}_T, g(x) \leq a\}$, on montre facilement que les fonctions caractéristiques de ces deux ensembles sont liées par la relation :

$$\chi_{\mathcal{X}_T^{\text{ad}}}\left(\boldsymbol{X}\right) = \mathbb{E}\left(\chi_{\mathbb{X}_T^{\text{ad}}}\left(\boldsymbol{X}\right)\right).$$

La contrainte presque-sûre peut donc être incorporée dans le critère du problème (5.3) comme une espérance portant sur l'état final qui s'ajoute au coût final $K(\boldsymbol{X}_T)$. On reste dans le cadre des problèmes étudiés au §5.2.2, et le problème peut être résolu par programmation dynamique en remplaçant $K(\cdot)$ par $K(\cdot) + \chi_{\mathbb{X}_T^{\text{ad}}}(\cdot)$. Il y a donc consistance dynamique.

Intéressons-nous maintenant à l'ajout au problème (5.3) d'une contrainte en probabilité portant sur le dernier pas de temps T. On souhaite par exemple amener le stock dans une certaine configuration au dernier pas de temps avec une probabilité d'au plus a :

$$\mathbb{P}\left(h\left(\boldsymbol{X}_T\right) \geq b\right) \leq a.$$

Remarquons qu'une telle contrainte peut toujours être modélisée comme une contrainte en espérance de la manière suivante :

$$\mathbb{E}\left(\mathbf{1}_{\{h(\boldsymbol{X}_T)\geq b\}}\right) \leq a,$$

avec $\mathbf{1}_A$ la fonction indicatrice associée à l'ensemble A. Il faut cependant noter que les contraintes en probabilité apportent des difficultés supplémentaires importantes, tant sur le plan théorique que sur le plan numérique, notamment autour des questions de connexité et de convexité de l'ensemble admissible qu'elles produisent, et ce même dans un cadre statique. On pourra consulter l'ouvrage de Prékopa (1995). Une partie du recueil de Ruszczyński et Shapiro (2003, Chapitre 5) est également consacrée à ce type de contraintes (voir aussi Henrion, 2002, Henrion et Strugarek, 2008, sur le sujet). Nous ne discuterons pas de ces questions ici. La difficulté qui nous intéresse est en effet commune aux contraintes en probabilité et aux contraintes en espérance. C'est la raison pour laquelle nous nous concentrons par la suite sur l'ajout au problème (5.3) de la contrainte suivante.

$$\mathbb{E}\left(g\left(\boldsymbol{X}_T\right)\right) \leq a. \tag{5.6}$$

Le lecteur familier des contraintes en probabilité pourra voir le niveau a comme un niveau de probabilité qu'un évènement doit satisfaire au dernier pas de temps.

4. Nous ne considérerons que des contraintes portant sur le dernier instant (voir Remarque 5.3).

Remarque 5.3 (Contraintes en probabilité jointes). Plutôt que considérer une contrainte en probabilité sur l'état final du type $\mathbb{P}\left(h\left(\boldsymbol{X}_T\right) \geq b\right) \leq a$, on peut vouloir s'intéresser à une contrainte plus générale, de la forme :

$$\mathbb{P}\left(h_t\left(\boldsymbol{X}_t\right) \geq b_t, \forall t = t_1, \ldots, T\right) \leq a.$$

Cette dernière peut en fait être modélisée comme la précédente, via l'introduction d'une variable aléatoire supplémentaire à valeurs binaires, dont la dynamique s'écrit :

$$\begin{aligned} \boldsymbol{Y}_{t_0} &= 1, \\ \boldsymbol{Y}_{t+1} &= \boldsymbol{Y}_t \times \mathbf{1}_{\{h_{t+1}(\boldsymbol{X}_{t+1}) \geq b_{t+1}\}}, \qquad \forall t = t_0, \ldots, T-1. \end{aligned}$$

Elle s'écrit alors : $\mathbb{E}\left(\boldsymbol{Y}_T\right) \leq a$.

Les problèmes ultérieurs, formulés à partir d'un temps initial $t_i > t_0$ sont déduits de manière naturelle du problème initial (par troncature de la dynamique et de la fonction coût). On choisit de garder le niveau a inchangé pour tous les problèmes suivants. Il faut être conscient qu'il s'agit d'un choix important (et probablement naïf) de modélisation. Il influera directement sur la propriété de consistance dynamique. Nous en discutons dans §5.3. Pour rappel, les problèmes qui nous intéressent à présent, indexés par le temps t_i s'écrivent donc :

$$\min_{\boldsymbol{X},\boldsymbol{U}} \quad \mathbb{E}\left(\sum_{t=t_i}^{T-1} L_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right) + K\left(\boldsymbol{X}_T\right)\right),$$

sous les contraintes de dynamique :

$$\begin{aligned} &\boldsymbol{X}_{t_i} \text{ donné}, \\ &\boldsymbol{X}_{t+1} = f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right), \qquad \forall t = t_i, \ldots, T-1, \end{aligned}$$

la contrainte de non-anticipativité :

$$\boldsymbol{U}_t \preceq \boldsymbol{X}_{t_i}, \boldsymbol{W}_{t_{i+1}}, \ldots, \boldsymbol{W}_t, \qquad \forall t = t_i, \ldots, T-1,$$

et la contrainte en espérance :

$$\mathbb{E}\left(g\left(\boldsymbol{X}_T\right)\right) \leq a.$$

Supposons qu'il existe une solution au problème au premier instant. Comme auparavant, nous sommes à la recherche, à chaque pas de temps t, de stratégies $\Phi_{t_0,t}^\sharp$ dépendant de la variable $\boldsymbol{X}_t$. Le premier indice t_0 se réfère au pas de temps à partir duquel le problème est posé, alors que le second indice se réfère au pas de temps auquel la décision est prise.

Il faut être conscient du fait que les solutions dépendent maintenant de la condition initiale $\boldsymbol{X}_{t_0}$ en tant que variable aléatoire. En effet, soit μ_T la loi de probabilité de $\boldsymbol{X}_T$. La contrainte (5.6) peut aussi s'écrire $\langle g, \mu_T \rangle \leq a$, de sorte que, dans la formulation distribuée, le problème à l'instant initial s'énonce :

$$\min_{\Phi,\mu} \quad \sum_{t=t_0}^{T-1} \left\langle \Lambda_t^{\Phi_t}, \mu_t \right\rangle + \langle K, \mu_T \rangle,$$

sous les contraintes dynamiques :

$$\begin{aligned} &\mu_{t_0} \text{ donnée,} \\ &\mu_{t+1} = \left(A_t^{\Phi_t}\right)^\star \mu_t, \qquad \forall t = t_0, \dots, T-1, \end{aligned}$$

et la contrainte finale :

$$\langle g, \mu_T \rangle \leq a.$$

Bien que ce problème semble linéaire par rapport aux variables μ_t, la dernière contrainte introduit une importante différence par rapport au cas précédent. Cette contrainte se met dans la fonction coût sous la forme :

$$\chi_{\{\langle g, \mu_T \rangle \leq a\}}.$$

La dynamique est toujours linéaire en les μ_t, $t = t_0, \dots, T$, mais la fonction coût n'est plus linéaire en μ_T, et donc plus linéaire en la condition initiale μ_{t_0}. Il n'y a donc aucune raison que les *feedbacks* ne dépendent pas de la condition initiale, comme c'était le cas au §5.2.2.

Nous faisons maintenant une remarque relative à la condition initiale. La structure d'information que nous considérons ici correspond à l'observation parfaite de $\boldsymbol{X}_t$ à l'instant t. Ainsi, la condition initiale naturelle est en fait déterministe :

$$\boldsymbol{X}_{t_0} = x_{t_0},$$

avec x_{t_0} la valeur observée du stock à l'instant t_0. Alors la loi de probabilité de $\boldsymbol{X}_{t_0}$ est la distribution de Dirac $\delta_{x_{t_0}}$.[5] Le raisonnement fait à l'instant t_0 est vrai pour les problèmes suivants commençant à l'instant t_i : dès lors que l'observation x_{t_i} de la variable aléatoire $\boldsymbol{X}_{t_i}$ est disponible, il est naturel d'écrire le problème d'optimisation (5.4) avec comme condition initiale :

$$\boldsymbol{X}_{t_i} = x_{t_i}.$$

Autrement dit, la loi de probabilité initiale dans chacun des problèmes successifs a toutes les raisons d'être une distribution de Dirac. La dynamique de telles conditions initiales est à rapprocher de celle du filtre dégénéré correspondant à un modèle d'information parfaite. Dans la suite, nous supposons que la condition initiale de chaque problème est donnée par une distribution de Dirac.

Selon le lemme 5.2, les problèmes d'optimisation formulés aux pas de temps ultérieurs, disons t_i, sont consistants dynamiquement si leur condition initiale est donnée par la loi provenant de l'équation de Fokker-Planck optimale :

$$\mu^\sharp_{t_0,t_i} = \left(A_{t_{i-1}}^{\Phi^\sharp_{t_0,t_{i-1}}}\right)^\star \dots \left(A_{t_0}^{\Phi^\sharp_{t_0,t_0}}\right)^\star \mu_{t_0}.$$

Cependant, à l'exception de problèmes où les bruits seraient en fait identiquement nuls, une telle distribution $\mu^\sharp_{t_0,t_i}$ est toujours différente d'une distribution de Dirac qui est,

5. La loi initiale μ_{t_0} du problème (5.3) se rapporte à l'information disponible sur $\boldsymbol{X}_{t_0}$ *avant* que $\boldsymbol{X}_{t_0}$ ne soit observé. Il paraît cependant plus raisonnable en pratique d'utiliser toute l'information disponible lorsque nous écrivons le problème à chaque nouveau temps initial, et ainsi d'utiliser une distribution de Dirac comme condition initiale.

comme nous venons de l'expliquer, la condition initiale naturelle avec laquelle reformuler le problème à l'instant t_i. Ainsi, la suite de problèmes n'est pas consistante dynamiquement si on considère des *feedbacks* Φ_t dépendant uniquement de $\boldsymbol{X}_t$.

Dans l'esprit du lemme 5.3, on propose alors de résoudre le problème déterministe par la programmation dynamique. En utilisant les notations de la formulation distribuée du problème de commande optimale stochastique, on peut écrire l'équation de la programmation dynamique portant sur les lois de probabilités :

$$V_T(\mu) = \begin{cases} \langle K, \mu \rangle & \text{si } \langle g, \mu \rangle \leq a, \\ +\infty & \text{sinon,} \end{cases}$$

pour toute loi de probabilité μ sur $\mathcal{X}_T$ et, pour tout $t = t_0, \dots, T-1$ et toute loi de probabilité μ sur $\mathcal{X}_t$:

$$V_t(\mu) = \min_{\Phi_t} \left\langle \Lambda_t^{\Phi_t}, \mu \right\rangle + V_{t+1}\left(\left(A_t^{\Phi_t}\right)^\star \mu\right).$$

Le cadre est similaire à celui de l'exemple déterministe du §5.2.1, et le lemme 5.3 nous indique que la suite de problèmes est consistante dynamiquement. Pour le problème qui nous intéresse ici, on obtient donc des stratégies $\Gamma_t : \mu_t \to \Phi_t$ en *feedback* sur les lois de probabilité μ_t. Autrement dit, la famille de problèmes sous contrainte en espérance est consistante dynamiquement dès lors que l'on cherche les stratégies comme des fonctions de la variable $\boldsymbol{X}_t$ *et* de la loi de probabilité de $\boldsymbol{X}_t$.

Cette équation de la programmation dynamique est bien sûr seulement conceptuelle. La résolution d'une telle équation n'est pas possible en pratique puisque la variable d'état est la loi de probabilité μ_t, qui est un objet de dimension infinie.

5.3 Retour à la dimension finie

Dans l'exemple précédent, on vient de montrer que le calcul de *feedbacks* sur le couple (x, μ) permettait de retrouver la consistance dynamique. Cela dit, rien ne nous indique qu'il s'agit de la statistique minimale ayant les propriétés recherchées. Effectivement, nous montrons maintenant qu'il existe un état de plus petite taille (et même de dimension finie) pour le problème.

5.3.1 Problème équivalent

Nous reprenons dans la cadre du temps discret les résultats de Bouchard, Elie, et Touzi (2009), qui proposent de remplacer, via l'ajout d'une commande et d'un état supplémentaires, la contrainte en espérance (5.6) par une contrainte presque-sûre. Afin de simplifier les notations, notons $\mathcal{A}_t$ la tribu engendrée par le passé des bruits jusqu'à l'instant t. On pose alors le problème de commande optimale stochastique suivant.

$$\min_{\boldsymbol{X},\boldsymbol{Z},\boldsymbol{U},\boldsymbol{V}} \quad \mathbb{E}\left(\sum_{t=t_0}^{T-1} L_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right) + K\left(\boldsymbol{X}_T\right)\right), \tag{5.7a}$$

sous les contraintes dynamiques sur $\boldsymbol{X}$:

$$\boldsymbol{X}_{t_0} = x_{t_0}, \tag{5.7b}$$
$$\boldsymbol{X}_{t+1} = f_t\left(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{W}_{t+1}\right), \qquad \forall t = t_0, \dots, T-1, \tag{5.7c}$$

les contraintes dynamiques sur $\boldsymbol{Z}$:

$$\boldsymbol{Z}_{t_0} = 0, \tag{5.7d}$$
$$\boldsymbol{Z}_{t+1} = \boldsymbol{Z}_t + \boldsymbol{V}_t, \qquad \forall t = t_0, \dots, T-1, \tag{5.7e}$$

les contraintes de non-anticipativité :

$$\boldsymbol{U}_t \preceq \mathcal{A}_t, \qquad \boldsymbol{V}_t \preceq \mathcal{A}_{t+1}, \qquad \forall t = t_0, \dots, T-1, \tag{5.7f}$$

la contrainte presque-sûre portant sur l'instant final :

$$g\left(\boldsymbol{X}_T\right) - \boldsymbol{Z}_T \leq a, \qquad \mathbb{P}\text{-p.s.}, \tag{5.7g}$$

et les contraintes intermédiaires sur les nouvelles commandes :

$$\mathbb{E}\left(\boldsymbol{V}_t \mid \mathcal{A}_t\right) = 0, \qquad \forall t = t_0, \dots, T-1. \tag{5.7h}$$

Par rapport au problème précédent, nous avons donc un nouvel état $(\boldsymbol{X}_t, \boldsymbol{Z}_t)$, ainsi qu'une nouvelle commande $\boldsymbol{V}_t$ qui, contrairement à $\boldsymbol{U}_t$, est en *Hasard-Décision*. Nous avons alors le théorème suivant.

Théorème 5.5. *Les problèmes* (5.3) *et* (5.7) *sont équivalents.*

Démonstration. Montrons que les ensembles admissibles de ces deux problèmes contiennent les mêmes processus de commande $(\boldsymbol{U}_{t_0}, \dots, \boldsymbol{U}_{T-1})$.

(5.3) $\subset$ (5.7) Soient des processus $(\boldsymbol{U}_{t_0}, \dots, \boldsymbol{U}_{T-1})$ et $(\boldsymbol{X}_{t_0}, \dots, \boldsymbol{X}_T)$ satisfaisant les contraintes du problème (5.3). On définit alors les processus $\boldsymbol{Z}$ et $\boldsymbol{V}$ par :

$$\begin{aligned}
\boldsymbol{Z}_{t_0} &= 0, \\
\boldsymbol{Z}_{t+1} &= \boldsymbol{Z}_t + \boldsymbol{V}_t, \qquad \forall t = t_0, \dots, T-1, \\
\boldsymbol{V}_t &= \mathbb{E}\Big(g\left(\boldsymbol{X}_T\right) \;\Big|\; \mathcal{A}_{t+1}\Big) - \mathbb{E}\Big(g\left(\boldsymbol{X}_T\right) \;\Big|\; \mathcal{A}_t\Big), \qquad \forall t = t_0, \dots, T-1.
\end{aligned}$$

Par sommation, on obtient la relation :

$$\boldsymbol{Z}_T = g\left(\boldsymbol{X}_T\right) - \mathbb{E}\Big(g\left(\boldsymbol{X}_T\right) \;\Big|\; \mathcal{A}_{t_0}\Big) = g\left(\boldsymbol{X}_T\right) - \mathbb{E}\Big(g\left(\boldsymbol{X}_T\right)\Big) \geq g\left(\boldsymbol{X}_T\right) - a.$$

Les processus $\boldsymbol{X}$, $\boldsymbol{Z}$, $\boldsymbol{U}$ et $\boldsymbol{V}$ vérifient donc bien les contraintes du problème (5.7).

(5.7) $\subset$ (5.3) Soient des processus $\boldsymbol{X}$, $\boldsymbol{Z}$, $\boldsymbol{U}$ et $\boldsymbol{V}$ vérifiant les contraintes du problème d'optimisation (5.7). On remarque que $\boldsymbol{Z}_T = \sum_{t=t_0}^{T-1} \boldsymbol{V}_t$, d'où l'on a que :

$$\mathbb{E}\left(\boldsymbol{Z}_T\right) = \sum_{t=t_0}^{T-1} \mathbb{E}\Big(\mathbb{E}\left(\boldsymbol{V}_t \mid \mathcal{A}_t\right)\Big) = 0.$$

Comme, par hypothèse, on a que $g(\boldsymbol{X}_T) - \boldsymbol{Z}_T \leq a$, on en déduit que $\mathbb{E}\left(g(\boldsymbol{X}_T)\right) \leq a$, ce qui montre que les processus $\boldsymbol{X}$ et $\boldsymbol{U}$ vérifient les contraintes du problème (5.3).

Les deux problèmes ont même ensemble admissible et même critère, d'où l'équivalence. □

Nous pouvons maintenant interpréter les nouvelles variables $\boldsymbol{V}_t$ et $\boldsymbol{Z}_t$. Plus $\boldsymbol{Z}_T$ est grand, plus la contrainte finale (5.7g) est facile à satisfaire. Cette variable permet de "jouer" sur la distribution de la variable aléatoire $g(\boldsymbol{X}_T) - a$. Ainsi, il sera économiquement intéressant pour le décideur de choisir une valeur de $\boldsymbol{Z}_T$ grande pour les scénarios "difficiles". Les contraintes (5.7e) et (5.7h), à l'aide de la variable de commande $\boldsymbol{V}_t$, permettent d'assurer que l'équilibre, c'est-à-dire la contrainte en espérance (5.6) du problème de départ, soit satisfaite. Autrement dit, si on choisit effectivement une valeur de $\boldsymbol{Z}_T$ grande pour les scénarios "difficiles", il nous faut "rééquilibrer" en imposant une valeur de $\boldsymbol{Z}_T$ petite pour un autre paquet de scénarios.

5.3.2 Principe de programmation dynamique

Il s'agit maintenant de résoudre le problème (5.7) par programmation dynamique avec un état de dimension finie. L'application du principe de programmation dynamique à ce problème n'est pas directe ; la raison principale est l'ajout des contraintes (5.7h). De plus, la mesurabilité des différentes variables de commande varie : l'une est mesurable par rapport à $\mathcal{A}_t$ (on dit qu'elle est en *Décision-Hasard*) alors que l'autre est mesurable par rapport à $\mathcal{A}_{t+1}$ (on dit qu'elle est en *Hasard-Décision*). Il convient d'expliquer comment s'obtient l'équation de la programmation dynamique dans ce cadre.

Dans la suite, on s'intéressera essentiellement à faire des *considérations structurelles* sur les problèmes étudiés, et on se préoccupera peu des *résultats de mesurabilité*. Les résultats fournis par l'analyse structurelle sont ceux que l'on peut obtenir pourvu que la mesurabilité des fonctions mises en jeu dans l'étude ne soit pas problématique : c'est par exemple le cas lorsque l'espace de probabilité sur lequel on travaille est fini ou dénombrable. L'analyse de la mesurabilité dans le cas général est une tâche complexe, qui ne sera pas regardée en détail, dont on trouvera des éléments importants d'analyse dans l'ouvrage de Bertsekas et Shreve (1996).

Cas "Décision-Hasard"

Oublions un instant les nouvelles variables $\boldsymbol{Z}$ et $\boldsymbol{V}$ et plaçons-nous dans le cadre du problème (5.3) où les commandes sont en *Décision-Hasard*. Pour obtenir l'équation de la programmation dynamique pour ce problème, on commence par faire apparaître la filtration $\mathcal{A} = (\mathcal{A}_t)_{t=t_0,\dots,T}$ dans le critère de la manière suivante.

$$\mathbb{E}\left(L_{t_0}\left(\boldsymbol{X}_{t_0}, \boldsymbol{U}_{t_0}, \boldsymbol{W}_{t_1}\right) + \dots \mathbb{E}\left(L_{T-1}\left(\boldsymbol{X}_{T-1}, \boldsymbol{U}_{T-1}, \boldsymbol{W}_T\right) + K\left(\boldsymbol{X}_T\right) \mid \mathcal{A}_{T-1}\right) \dots \mid \mathcal{A}_{t_0}\right)$$

Comme $\boldsymbol{U}_t$ est mesurable par rapport à $\mathcal{A}_t$ et que les équations dynamiques font qu'il en est de même pour $\boldsymbol{X}_t$, on fait rentrer les opérateurs de minimisation "aussi loin que possible" sous les espérances conditionnelles. Enfin, on effectue ces opérations de minimisation en commençant par la plus intérieure, c'est-à-dire celle portant sur la commande $\boldsymbol{U}_{T-1}$. Elle s'écrit :

$$\begin{aligned} \min_{\boldsymbol{U}_{T-1} \preceq \mathcal{A}_{T-1}} \quad & \mathbb{E}\left(\min_{\boldsymbol{X}_T \preceq \mathcal{A}_T} L_{T-1}\left(\boldsymbol{X}_{T-1}, \boldsymbol{U}_{T-1}, \boldsymbol{W}_T\right) + K\left(\boldsymbol{X}_T\right) \;\middle|\; \mathcal{A}_{T-1}\right), \\ \text{s.c.} \quad & \boldsymbol{X}_T = f_{T-1}\left(\boldsymbol{X}_{T-1}, \boldsymbol{U}_{T-1}, \boldsymbol{W}_T\right), \end{aligned}$$

problème qui s'écrit plus simplement :

$$\min_{\boldsymbol{U}_{T-1} \preceq \mathcal{A}_{T-1}} \quad \mathbb{E}\left(L_{T-1}\left(\boldsymbol{X}_{T-1}, \boldsymbol{U}_{T-1}, \boldsymbol{W}_T\right) + K\left(f_{T-1}\left(\boldsymbol{X}_{T-1}, \boldsymbol{U}_{T-1}, \boldsymbol{W}_T\right)\right) \mid \mathcal{A}_{T-1}\right). \quad (5.8)$$

Les variables $\boldsymbol{U}_{T-1}$ et $\boldsymbol{X}_{T-1}$ sont $\mathcal{A}_{T-1}$-mesurables. De plus, le bruit $\boldsymbol{W}_T$ est indépendant de la tribu $\mathcal{A}_{T-1}$. On a donc que l'espérance conditionnelle de (5.8) se réduit à une simple espérance sur $\boldsymbol{W}_T$, qui produit une fonction $\mathcal{C}(\boldsymbol{X}_{T-1}, \boldsymbol{U}_{T-1})$. La minimisation en $\boldsymbol{U}_{T-1}$ se fait alors paramétriquement par rapport aux valeurs que prend $\boldsymbol{X}_{T-1}$, et produit ainsi une solution sous la forme d'un *feedback* $\Phi^\sharp_{T-1}(\boldsymbol{X}_{T-1})$ et une fonction coût optimal [6] :

$$V^\sharp_{T-1}(\boldsymbol{X}_{T-1}) := \min_{\boldsymbol{U}_{T-1} \preceq \mathcal{A}_{T-1}} \mathcal{C}(\boldsymbol{X}_{T-1}, \boldsymbol{U}_{T-1}),$$

qui correspond à la fonction valeur (au sens de la programmation dynamique) du problème à l'instant $T-1$.

Le problème à l'instant $T-2$ se formule alors de manière semblable, la fonction $V^\sharp_{T-1}$ prenant la place de la fonction K. Par récurrence, on obtient les équations de la programmation dynamique.

Cas "Hasard-Décision"

Compliquons un peu le problème et plaçons-nous dans le cas *Hasard-Décision* où la contrainte de mesurabilité sur la commande, qui s'écrivait $\boldsymbol{U}_t \preceq \mathcal{A}_t$, est remplacée par la contrainte $\boldsymbol{U}_t \preceq \mathcal{A}_{t+1}$. Ainsi, au moment de prendre la décision à l'instant t, on connaît la réalisation du bruit $\boldsymbol{W}_{t+1}$. Les opérateurs de minimisation en $\boldsymbol{U}_t$ rentrent "un cran plus loin" sous l'espérance, et le problème d'optimisation correspondant à l'instant $T-1$ devient :

$$\mathbb{E}\left(\min_{\boldsymbol{U}_{T-1} \preceq \mathcal{A}_T} \quad L_{T-1}(\boldsymbol{X}_{T-1}, \boldsymbol{U}_{T-1}, \boldsymbol{W}_T) + K\left(f_{T-1}(\boldsymbol{X}_{T-1}, \boldsymbol{U}_{T-1}, \boldsymbol{W}_T)\right) \;\middle|\; \mathcal{A}_{T-1} \right).$$

La minimisation en $\boldsymbol{U}_{T-1}$ sous l'espérance s'effectue paramétriquement par rapport aux valeurs prises par $\boldsymbol{X}_{T-1}$ et $\boldsymbol{W}_T$ et conduit à une stratégie optimale sous la forme d'un *feedback* $\Phi^\sharp_{T-1}(\boldsymbol{X}_{T-1}, \boldsymbol{W}_T)$. Puis, le calcul d'espérance conditionnelle se réduit à un calcul d'espérance portant sur le bruit $\boldsymbol{W}_T$, produisant ainsi une fonction valeur $V^\sharp_{T-1}(\boldsymbol{X}_{T-1})$. De la même manière que dans le cas précédent, on obtient par récurrence les équations de la programmation dynamique.

Cas "Hasard-Décision" avec contrainte

Compliquons encore un peu plus et ajoutons à notre problème en *Hasard-Décision* une contrainte du type (5.7h) :

$$\mathbb{E}(\boldsymbol{U}_t \mid \mathcal{A}_t) = 0,$$

à chaque instant t. Remarquons d'abord que les raisonnements faits précédemment restent inchangés jusqu'au stade où l'on obtient le problème de minimisation à l'instant $T-1$ qui s'écrit maintenant :

$$\min_{\boldsymbol{U}_{T-1} \preceq \mathcal{A}_T} \quad \mathbb{E}\left(L_{T-1}(\boldsymbol{X}_{T-1}, \boldsymbol{U}_{T-1}, \boldsymbol{W}_T) + K\left(f_{T-1}(\boldsymbol{X}_{T-1}, \boldsymbol{U}_{T-1}, \boldsymbol{W}_T)\right) \mid \mathcal{A}_{T-1}\right),$$

sous la contrainte couplante :

$$\mathbb{E}(\boldsymbol{U}_{T-1} \mid \mathcal{A}_{T-1}) = 0.$$

6. On utilise ici un théorème de sélection mesurable implicite afin de pouvoir dire que la stratégie optimale et que la fonction valeur sont mesurables par rapport à $\boldsymbol{X}_{T-1}$. Le lecteur pourra se référer au travail de Carpentier *et al.* (2009b), parmi d'autres, pour l'utilisation d'un argument de cette nature.

Peut-on encore affirmer que la solution $\boldsymbol{U}_{T-1}^\sharp$ à ce problème est mesurable par rapport au couple $(\boldsymbol{X}_{T-1}, \boldsymbol{W}_T)$? Nous donnons la réponse à cette question dans le cadre épuré suivant. On se donne une variable aléatoire $\boldsymbol{Y}$, une variable aléatoire $\boldsymbol{X}$ *mesurable* par rapport à $\boldsymbol{Y}$ et une variable aléatoire $\boldsymbol{W}$ *indépendante* de $\boldsymbol{Y}$. On note $\mathcal{G}$ la tribu engendrée par $\boldsymbol{Y}$ et $\mathcal{F}$ la tribu engendrée par $(\boldsymbol{Y}, \boldsymbol{W})$ (et donc $\mathcal{F} = \mathcal{G} \vee \sigma(\boldsymbol{W})$). On s'intéresse au problème suivant :

$$\min_{\boldsymbol{U} \preceq \mathcal{F}} \mathbb{E}\Big(j(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{W}) \;\Big|\; \mathcal{G}\Big) \quad \text{sous} \quad \mathbb{E}\Big(\boldsymbol{U} \;\Big|\; \mathcal{G}\Big) = 0. \tag{5.9}$$

Théorème 5.6. *Il existe une solution $\boldsymbol{U}^\sharp$ du problème* (5.9) *telle que $\boldsymbol{U}^\sharp$ soit mesurable par rapport au couple $(\boldsymbol{X}, \boldsymbol{W})$.*

Démonstration. On donne la preuve de ce théorème dans le cas où la variable aléatoire $\boldsymbol{Y}$ est finie ; on note $y_1, \ldots, y_N$ les valeurs que prend $\boldsymbol{Y}$. Le problème (5.9) est alors formé d'une famille de N sous-problèmes d'optimisation indépendants, dont la i-ème instance correspond au conditionnement par rapport à $\{\boldsymbol{Y} = y_i\}$:

$$\min_{\boldsymbol{U} \preceq (\boldsymbol{Y}, \boldsymbol{W})} \mathbb{E}\Big(j(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{W}) \;\Big|\; \boldsymbol{Y} = y_i\Big) \quad \text{sous} \quad \mathbb{E}\Big(\boldsymbol{U} \;\Big|\; \boldsymbol{Y} = y_i\Big) = 0. \tag{5.10}$$

La variable aléatoire $\boldsymbol{X}$ étant mesurable par rapport à $\boldsymbol{Y}$, il existe une application mesurable ϑ (voir Dellacherie et Meyer, 1975, Chapitre 1, p. 18, pour l'existence d'une telle fonction p), telle que l'on ait : $\boldsymbol{X} = \vartheta(\boldsymbol{Y})$; on note $x_i = \vartheta(y_i)$. De plus, la mesurabilité de $\boldsymbol{U}$ par rapport à $(\boldsymbol{Y}, \boldsymbol{W})$ fait que $\boldsymbol{U}$ se met sous la forme :

$$\boldsymbol{U} = \sum_{i=1}^{N} \boldsymbol{U}_i \mathbf{1}_{\{\boldsymbol{Y} = y_i\}},$$

où chaque $\boldsymbol{U}_i$ est mesurable par rapport à $\boldsymbol{W}$. On en déduit que le problème (5.10) se met sous la forme : [7]

$$\min_{\boldsymbol{U}_i \preceq \boldsymbol{W}} \mathbb{E}\Big(j(x_i, \boldsymbol{U}_i, \boldsymbol{W})\Big) \quad \text{sous} \quad \mathbb{E}\Big(\boldsymbol{U}_i\Big) = 0.$$

La solution $\boldsymbol{U}_i^\sharp$ de ce dernier problème, d'une part est mesurable par rapport à $\boldsymbol{W}$, et d'autre part dépend paramétriquement de la valeur x_i (plutôt que de y_i). On en déduit que toute solution $\boldsymbol{U}^\sharp$ du problème (5.9) se met sous la forme :

$$\boldsymbol{U}^\sharp = \sum_{i=1}^{N} \boldsymbol{U}_i^\sharp \mathbf{1}_{\{\boldsymbol{X} = x_i\}},$$

et donc que $\boldsymbol{U}^\sharp$ est mesurable par rapport à $(\boldsymbol{X}, \boldsymbol{W})$. □

Remarque 5.4. Il faut bien se rendre compte que la présence de la variable $\boldsymbol{W}$ dans le *feedback* optimal n'est pas due à la présence de $\boldsymbol{W}$ dans le critère du problème (5.9), mais correspond au fait que $\boldsymbol{W}$ est la variable aléatoire qu'il faut "ajouter" au conditionnement $\mathcal{G}$ du problème pour obtenir la tribu $\mathcal{F}$ imposant la mesurabilité de la commande $(\mathcal{F} = \mathcal{G} \vee \sigma(\boldsymbol{W}))$. Cet effet est appelé le *saut de tribu*. Pour se convaincre du phénomène, on considère le problème :

$$\min_{\boldsymbol{U} \preceq \mathcal{F}} \mathbb{E}\Big(j(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{W}) \;\Big|\; \boldsymbol{X}\Big) \quad \text{sous} \quad \mathbb{E}\Big(\boldsymbol{U} \;\Big|\; \boldsymbol{X}\Big) = 0,$$

7. Les espérances conditionnelles deviennent des espérances car $\boldsymbol{W}$ est indépendante de $\boldsymbol{Y}$.

et l'on suppose que la tribu $\mathcal{H}$ engendrée par $\boldsymbol{X}$ est telle que l'on ait : $\mathcal{G} = \mathcal{H} \vee \sigma(\widetilde{\boldsymbol{W}})$, où $\widetilde{\boldsymbol{W}}$ est une variable aléatoire indépendante de $\boldsymbol{X}$. Alors, la démonstration du théorème précédent montre que la solution de ce problème est en *feedback* a priori sur $(\boldsymbol{X}, \widetilde{\boldsymbol{W}}, \boldsymbol{W})$ et non sur seulement $(\boldsymbol{X}, \boldsymbol{W})$. On peut construire, dans le cas non convexe, des exemples où la dépendance du *feedback* en $\widetilde{\boldsymbol{W}}$ est effective. [8]

Cas mixte

Des résultats précédents, on déduit que l'on peut utiliser la technique de programmation dynamique pour résoudre le problème (5.7) (pour lequel, on le rappelle, l'état est de dimension finie). On est alors en présence de commandes en *Décision-Hasard*, ainsi que d'autres commandes en *Hasard-Décision* sur laquelle porte des contraintes intégrales. La relation de récurrence permettant de calculer les fonctions de Bellman prend la forme suivante :

– la fonction de Bellman est initialisée à l'instant T par :

$$V_T(x, z) = K(x) + \chi_{G^{\mathrm{ad}}}(x, z),$$

où $\chi_{G^{\mathrm{ad}}}$ est la fonction caractéristique de l'ensemble $G^{\mathrm{ad}} = \Big\{(x, z) \mid g(x) - z \leq a\Big\}$;

– la fonction de Bellman à l'instant t est donnée par :

$$V_t(x, z) = \min_{u \in \mathbb{U}} \quad \min_{\boldsymbol{V} \preceq \boldsymbol{W}_{t+1}} \mathbb{E}\Big(L_t(x, u, \boldsymbol{W}_{t+1}) + V_{t+1}\Big(f_t(x, u, \boldsymbol{W}_{t+1}), z + \boldsymbol{V}\Big)\Big),$$

sous la contrainte intégrale :

$$\mathbb{E}\,(\boldsymbol{V}) = 0.$$

Le calcul de la fonction V_t fournit un contrôle optimal $\boldsymbol{U}_t^\sharp$ mesurable par rapport à $(\boldsymbol{X}_t, \boldsymbol{Z}_t)$ et un contrôle optimal $\boldsymbol{V}_t^\sharp$ mesurable par rapport à $(\boldsymbol{X}_t, \boldsymbol{Z}_t, \boldsymbol{W}_{t+1})$.

Comme on l'a déjà souligné, le contrôle optimal ainsi obtenu satisfait naturellement la propriété de consistance dynamique pour la famille des problèmes formulés à partir de (5.7) pour les différents instants t_0, car ce contrôle est indépendant de la loi de probabilité de l'état.

5.4 Conclusion

Dans le cadre de la commande optimale, nous avons établi un parallèle entre la notion de consistance dynamique d'une famille de problèmes de décision et le concept de variable d'état, autour du principe de programmation dynamique. Ainsi une suite de problèmes d'optimisation est-elle dynamiquement consistante si chacun des problèmes est indexé par un nombre suffisant de paramètres, qui constituent alors un état pour le système.

Comme il est bien connu, une suite de problèmes d'optimisation qui est consistant dynamiquement ne le reste pas nécessairement lorsque l'on rajoute telle ou telle contrainte de risque. Cependant, nous avons montré qu'il était souvent possible de retrouver la propriété de consistance dynamique au prix d'une augmentation de l'état et de la commande.

Il resterait à tester la résolution numérique de l'équation de programmation dynamique mise en évidence.

8. Par contre, dans le cas convexe, on montre facilement par des arguments de type dualité que le *feedback* optimal ne dépend que du couple $(\boldsymbol{X}, \boldsymbol{W})$, et pas de $\widetilde{\boldsymbol{W}}$.

Conclusion et perspectives

Et plus que l'air marin la douceur angevine.

Heureux qui comme Ulysse...
Joachim du Bellay (1522-1560)

Ce mémoire est le résultat d'un travail qui se place dans la continuité de nombreux travaux de recherche et de thèses réalisés au sein d'un groupe de travail nommé Systems and Optimization Working Group (SOWG), abrité par l'École des Ponts-ParisTech et dont les membres permanents sont Pierre Carpentier, de L'École Nationale Supérieure de Techniques Avancées (ENSTA-ParisTech), Jean-Philippe Chancelier, Guy Cohen et Michel De Lara de l'École des Ponts-ParisTech. Ces travaux de recherche sont, pour l'essentiel, le fruit d'une longue collaboration avec le département Optimisation, Simulation, Risques et Statistique (OSIRIS) d'EDF R&D.

Contributions du mémoire

Un problème de commande optimale stochastique consiste en la recherche de stratégies optimales de commande d'un système dynamique, influencé par des aléas exogènes.

- Une méthodologie bien connue pour ce type de problème consiste à discrétiser la structure aléatoire du problème à l'aide de chroniques arborescentes. On cherche ainsi à représenter la diffusion de l'aléa dans le temps. Nous mettons en lumière, dans une première partie, les raisons pour lesquelles cette méthodologie n'est pas adaptée à la recherche de stratégies pour des problèmes à plusieurs pas de temps. En effet, nous observons que le nombre de scénarios nécessaire à l'obtention d'une précision donnée croît de manière exponentielle avec l'horizon de temps du problème. Ce résultat corrobore celui de Shapiro (2006), qui considérait l'erreur en terme d'estimation de la valeur du problème. L'originalité de notre travail est de considérer l'erreur sur l'estimation des stratégies et de montrer, dans un deuxième temps, que le défaut des arbres de scénarios n'est pas à imputer à l'usage de scénarios mais plutôt à la discrétisation sous forme d'arbres d'aléas, qui ne semble pas être la bonne manière d'approcher des espérances conditionnelles. Ainsi, nous montrons que les méthodes particulaires ont un bien meilleur comportement asymptotique, même lorsque l'horizon de temps grandit. Cela dit, elles recontrent des difficultés face à la dimension de l'espace d'état qui font qu'elles ne peuvent être considérées comme la réponse à la problématique adressée dans cette thèse.
- Une des difficultés inhérentes aux problèmes de commande optimale stochastique, dans le cadre de la programmation dynamique, est celle de la dimension de l'espace d'état. Nous recherchons des stratégies de commande qui sont des fonctions

dont l'espace de départ est l'état du système. Ainsi, lorsque celui-ci est de grande taille, rien que la représentation de telles fonctions sur un ordinateur devient problématique. Nous proposons ici une manière de réduire l'information nécessaire à la construction d'une stratégie de commande raisonnable [9] pour une certaine classe de problèmes d'optimisation dits décomposables. À notre connaissance, les premiers travaux proposant d'adapter au cas stochastique les algorithmes de décomposition des grands systèmes bien connus en déterministe en approchant statistiquement le multiplicateur de Lagrange sont ceux de Strugarek (2006, Chapitre 5) et de Barty et Roy (2007). L'algorithme *Dual Approximate Dynamic Programming* (DADP) que nous introduisons ici généralise en quelque sorte ces travaux. Nous montrons de quelle manière le problème de départ est affecté par l'approximation et étudions la convergence de l'algorithme. Nous présentons enfin une application à un problème de gestion de production électrique.

– Dans le dernier chapitre, nous nous intéressons à une propriété structurelle des problèmes d'optimisation dynamique : la consistance dynamique. Cette propriété est bien connue des économistes et bénéficie d'un intérêt important au sein de la communauté *Stochastic Programming*, particulièrement depuis quelques années. L'originalité du travail présenté est de mettre en évidence le lien qui existe entre la notion de consistance dynamique d'une suite de problèmes de décision et le concept de variable d'état en commande optimale. On trouve dans la littérature récente plusieurs travaux [10] dont le but est de rechercher des mesures de risque permettant de conserver la structure d'état "naturelle" du problème de départ. Notre approche consiste plutôt à montrer que, pour une large classe de problèmes de commande optimale stochastique, la propriété de consistance dynamique peut être obtenue quitte à modifier la structure d'état du système.

Perspectives

– En ce qui concerne les méthodes particulaires, il reste un travail important à mener sur plusieurs fronts. Le premier est théorique : nous ne bénéficions pas encore de résultats de convergence pour cette méthode, bien que l'expérience nous indique leur bon comportement dans nombre de cas. Rappelons que cela est dû au fait que nous n'utilisons une expression du gradient, lors du déroulement de l'algorithme, qui n'est vraie qu'à l'optimum. Le deuxième est pratique : les méthodes particulaires consistent en une méthode de gradient ; il serait intéressant d'améliorer la performance de l'algorithme en s'inspirant des méthodes classiques de l'optimisation numérique. Enfin, il serait bon de réfléchir à un nouveau protocole d'évaluation de la vitesse de convergence des méthodes à base de scénarios, que nous ne sommes parvenus à évaluer que numériquement.
– La méthodologie que nous proposons pour décomposer des problèmes de commande optimale stochastique de grande taille ouvre un grand champ d'évolutions possibles. Nous nous sommes intéressés à des systèmes ayant une structure en marguerite car ils sont relativement simples et permettent de couvrir une large classe de problèmes : ils apparaissent notamment dans le cadre de la gestion d'un portefeuille physique et/ou financier, ce que nous avons étudié ici. Nous avons en tête de nous intéresser

9. Nous n'utilisons volontairement pas le terme "optimal" pour insister sur le fait qu'il s'agit d'une approximation (que nous quantifions) du problème de départ.

10. Nous pensons notamment à ceux de Ruszczyński.

à des modèles où les interactions entre sous-systèmes sont plus complexes. On peut penser en premier lieu aux structures en cascades, qui apparaissent notamment dans le cadre de la gestion d'une vallée hydraulique où chaque réservoir, lorsqu'il produit, déverse dans le réservoir suivant. Puis, on peut penser à une application à des systèmes en réseau, où chaque unité peut interagir avec ses voisines. Par ailleurs, il conviendrait de s'intéresser à des méthodologies permettant d'identifier efficacement la "bonne" variable d'information, sous une contrainte de budget par exemple (en s'imposant la taille maximale de l'état qu'on est prêt à traiter dans les sous-problèmes). Enfin, il s'agirait de tester la mise en œuvre de méthodes adaptées à l'optimisation non-lisse couplée avec l'algorithme DADP, que ce soit le Lagrangien augmenté linéarisé ou bien la méthode des faisceaux.

– Sur le sujet de la consistance dynamique, il reste un certain nombre de questions intéressantes à traiter. Du point de vue de la modélisation, il s'agit de se poser la question de la nécessité pour une suite de problèmes de décision d'être consistante dynamiquement. En effet, contrairement à des contraintes physiques que l'on incorpore dans le modèle car les stratégies en sortie devront être implantables dans la réalité, il n'est pas clair que la consistance dynamique soit une contrainte qu'un décideur ait à s'imposer. Du point de vue numérique, il nous reste à tester la mise en œuvre du modèle avec retour à la dimension finie, faite au §5.3. Ces considérations numériques font l'objet d'une partie du sujet de la thèse de Jean-Christophe Alais qui débute dans le cadre d'un partenariat entre le CERMICS et le département OSIRIS d'EDF R&D.

Annexe A

Optimisation

On rappelle ici seulement les résultats utiles aux travaux présentés dans ce mémoire. Le lecteur intéressé pourra consulter les ouvrages de Cohen (2000) et de Gilbert (2010) pour un cours complet sur l'optimisation différentiable et l'optimisation convexe.

Les résultats qui suivent sont issus de l'ouvrage de Ekeland et Temam (1999). Soient $\mathcal{U}$ et Λ deux espaces de Hilbert [1], et $\mathcal{A}$ et $\mathcal{B}$ deux sous-ensembles de ceux-ci, respectivement. On introduit de plus une fonction $L : \mathcal{U} \times \Lambda \to \mathbb{R}$. On décrit ici les liens existant entre le problème :

$$\inf_{u \in \mathcal{A}} \sup_{\lambda \in \mathcal{B}} L(u, \lambda), \tag{A.1}$$

dit problème primal, et le problème :

$$\sup_{\lambda \in \mathcal{B}} \inf_{u \in \mathcal{A}} L(u, \lambda),$$

dit problème dual. Ainsi, on appelle $\mathcal{U}$ l'espace primal et Λ l'espace dual.

Définition A.1. On dit que le couple $(\overline{u}, \overline{\lambda}) \in \mathcal{A} \times \mathcal{B}$ est une point-selle de L sur $\mathcal{A} \times \mathcal{B}$ si :

$$L(\overline{u}, \lambda) \leq L\left(\overline{u}, \overline{\lambda}\right) \leq L\left(u, \overline{\lambda}\right), \qquad \forall u \in \mathcal{A}, \forall \lambda \in \mathcal{B}.$$

On cherche à caractériser l'existence d'un point-selle pour la fonction L. On a la proposition suivante :

Proposition A.2. *Si :*

1. $\mathcal{A} \subset \mathcal{U}$ *est convexe, fermé, non vide,*
2. $\mathcal{B} \subset \Lambda$ *est convexe, fermé, non vide,*
3. $\forall u \in \mathcal{A}$, $\lambda \to L(u, \lambda)$ *est concave, semi-continue supérieurement,*
4. $\forall \lambda \in \mathcal{B}$, $u \to L(u, \lambda)$ *est convexe, semi-continue inférieurement,*
5. $\mathcal{A}$ *et* $\mathcal{B}$ *sont bornés,*

alors L possède au moins un point-selle sur $\mathcal{A} \times \mathcal{B}$ et :

$$\min_{u \in \mathcal{A}} \max_{\lambda \in \mathcal{B}} L(u, \lambda) = \max_{\lambda \in \mathcal{B}} \min_{u \in \mathcal{A}} L(u, \lambda).$$

1. Ces résultats se généralisent à des espaces de Banach en adaptant certaines notions (voir Ekeland et Temam, 1999), mais nous n'en avons pas besoin ici.

Il faut noter que l'on peut remplacer l'hypothèse "$\mathcal{A}$ est borné", qui est en soit assez restrictive, par :

$$\exists \lambda_0 \in \mathcal{B} \text{ tel que } \lim_{u \in \mathcal{A}, \|u\| \to +\infty} L(u, \lambda_0) = +\infty,$$

ainsi que l'hypothèse "$\mathcal{B}$ est borné" par :

$$\exists u_0 \in \mathcal{A} \text{ tel que } \lim_{\lambda \in \mathcal{B}, \|\lambda\| \to +\infty} L(u_0, \lambda) = +\infty.$$

La proposition que nous venons d'énoncer nous indique que les problèmes primal et dual ont même valeur, mais ne nous dit rien sur les minimiseurs et maximiseurs.

On suppose maintenant que la fonction L s'écrit :

$$L(u, \lambda) = J(u) + \langle \lambda, g(u) \rangle .$$

On définit l'algorithme d'Uzawa de la manière suivante. On se donne un point initial $\lambda_0 \in \mathcal{B}$. À chaque itération $n \geq 0$, on calcule u_n qui minimise $J(u) + \langle \lambda_n, g(u) \rangle$, puis on met à jour λ_n :

$$\lambda_{n+1} = \Pi_{\mathcal{B}} (\lambda_n + \rho_n g(u_n)),$$

avec ρ_n un scalaire positif. Le théorème suivant nous indique dans quels cas cet algorithme nous permet d'obtenir une solution du problème (A.1).

Théorème A.3 (Ekeland et Temam, 1999, Chapitre VII, Proposition 1.1)**.** *Si :*

1. $\mathcal{A} \subset \mathcal{U}$ *est convexe, fermé, non vide,*
2. $\mathcal{B} \subset \Lambda$ *est convexe, fermé, non vide,*
3. $\mathcal{B}$ *est borné ou bien* $\mathcal{A}$ *est borné,*
4. J *est Gâteaux-différentiable,*
5. $\langle J'(u) - J'(v), u - v \rangle \geq \alpha \|u - v\|^2, \alpha > 0, \forall (u, v) \in \mathcal{A} \times \mathcal{B},$
6. $\forall \lambda \in \mathcal{B}, v \to \langle \lambda, g(v) \rangle$ *est convexe et semi-continue inférieurement sur* $\mathcal{A}$*,*
7. g *est Lipschitzienne :* $\|g(u) - g(v)\| \leq c\|u - v\|,$

alors :

1. L *a au moins un point-selle* $(\overline{u}, \overline{\lambda})$*,*
2. $\overline{u}$ *est unique et est solution du problème* (A.1)*,*
3. *l'algorithme d'Uzawa converge, au sens où :*

$$u_n \to \overline{u} \text{ dans } \mathcal{U},$$

si le pas ρ_n *de l'algorithme est choisi tel que* $2\alpha\rho_n - c^2\rho_n^2 \geq \beta > 0.$

Annexe B

Probabilités

Nous rappelons ici brièvement quelques notions élémentaires de théorie de la mesure et de probabilités permettant de comprendre les concepts de mesurabilité des variables aléatoires. Pour un cours complet sur la théorie de la mesure et les probabilités, on pourra consulter l'ouvrage de Billingsley (1995).

Définition B.1 (Tribu, Espace mesurable, Ensemble mesurable)**.** On appelle *tribu* ou σ-algèbre sur un ensemble Ω un ensemble $\mathcal{A}$ non vide de parties de Ω, stable par passage au complémentaire et par union dénombrable. On dit alors que $(\Omega, \mathcal{A})$ est un *espace mesurable*. Enfin, une partie de Ω est dite mesurable si elle appartient à la tribu $\mathcal{A}$.

Les exemples classiques de tribus sont :
- la tribu dite *triviale* $\mathcal{A} = \mathcal{P}(\Omega)$, qui est l'ensemble des parties de Ω ;
- la tribu dite *grossière* $\mathcal{A} = \{\emptyset, \Omega\}$.

On complète souvent la définition d'un espace mesurable en lui ajoutant une mesure $\mathbb{P}$ sur la tribu $\mathcal{A}$ telle que $\mathbb{P}(\Omega) = 1$, que l'on appelle *mesure de probabilité*. On dit alors que le triplet $(\Omega, \mathcal{A}, \mathbb{P})$ est un *espace de probabilité* ou *espace probabilisé*. On introduit ensuite la notion de fonction mesurable.

Définition B.2 (Fonction mesurable)**.** Soient E et F des espaces mesurables munis respectivement d'une tribu $\mathcal{E}$ et $\mathcal{F}$. Une fonction f de E dans F sera dite fonction mesurable de $(E, \mathcal{E})$ dans $(F, \mathcal{F})$ si pour tout B appartenant à $\mathcal{F}$, son image réciproque $f^{-1}(B)$ appartient à E.

Une *variable aléatoire* est une fonction mesurable de $(\Omega, \mathcal{A})$ vers un espace mesurable $(E, \mathcal{E})$. On se sert souvent de la notion de tribu engendrée par une variable aléatoire :

Définition B.3 (Tribu image réciproque/Tribu engendrée)**.** Soit $(E, \mathcal{E})$ un espace mesurable et $\boldsymbol{X}$ une variable aléatoire de $(\Omega, \mathcal{A})$ dans $(E, \mathcal{E})$. L'ensemble défini par :

$$\boldsymbol{X}^{-1}(\mathcal{E}) := \left\{ \boldsymbol{X}^{-1}(P) \ ; P \in \mathcal{E} \right\}$$

est une tribu sur Ω. On l'appelle *tribu image réciproque* ou *tribu engendrée* par $\boldsymbol{X}$.

Définition B.4 (Intégrande normale)**.** Soit $(\Omega, \mathcal{A}, \mathbb{P})$ un espace de probabilité, $\boldsymbol{W}$ une variable aléatoire à valeurs dans $\mathbb{W}$, un espace métrique puni de la tribu $\mathcal{A}_{\mathbb{W}}$, qui est égale à $\mathcal{A}$ transportée par $\boldsymbol{W}$, $\mathbb{U}$ un espace de Hilbert muni de sa tribu borélienne $\mathcal{A}_{\mathbb{U}}$, et $f : \mathbb{U} \times \mathbb{W} \to \mathbb{R}$ une application. On dit que f est une *intégrande normale* si et seulement si :

1. f est $\mathcal{A}_{\mathbb{U}} \otimes \mathcal{A}_{\mathbb{W}}$-mesurable ;
2. pour $\mathbb{P}$-presque tout $\omega \in \Omega$, $f(\cdot, \boldsymbol{W}(\omega)) : \mathbb{U} \to \mathbb{R}$ est semi-continue inférieurement.

Bibliographie

P. Artzner, F. Delbaen, J. M. Eber et D. Heath : Coherent measures of risk. *Mathematical Finance*, 9(3), 1999.

P. Artzner, F. Delbaen, J.-M. Eber, D. Heath et H. Ku : Coherent multiperiod risk-adjusted values and Bellman's principle. *Annals of Operations Research*, 152(1):5–22, July 2007.

L. Bacaud, C. Lemaréchal, A. Renaud et C. A. Sagastizábal : Bundle methods in stochastic optimal power management : A disaggregated approach using preconditioner. *Computational Optimization and Applications*, 20(3):227–244, 2001.

K. Barty : *Contributions à la discrétisation des contraintes de mesurabilité pour les problèmes d'optimisation stochastique*. Thèse de doctorat, École Nationale des Ponts et Chaussées, 2004.

K. Barty, P. Carpentier, J.-P. Chancelier, G. Cohen, M. de Lara et T. Guilbaud : Dual effect free stochastic controls. *Annals of Operations Research*, 142:41–62, 2 2006.

K. Barty, P. Carpentier et P. Girardeau : Decomposition of large-scale stochastic optimal control problems. *RAIRO Operations Research*, 44(3):167–183, 7 2010.

K. Barty, P. Girardeau, J.-S. Roy et C. Strugarek : Application of kernel-based stochastic gradient algorithms to option pricing. *Monte Carlo Methods and Applications*, 14(2):99–127, 2008.

K. Barty et J.-S. Roy : Stochastic decomposition of multistage problems using dual linear decision rules. International Conference on Stochastic Programming XI, Vienna, 8 2007.

K. Barty, J.-S. Roy et C. Strugarek : Hilbert-valued perturbed subgradient algorithms. *Mathematics of Operations Research*, 32:551–562, 2007.

R. Bellman : *Dynamic Programming*. Princeton University Press, New Jersey, 1957.

R. Bellman et S. E. Dreyfus : Functional approximations and dynamic programming. *Math tables and other aides to computation*, 13:247–251, 1959.

D. P. Bertsekas : *Constrained optimization and Lagrange multiplier methods*. Academic Press, 1982.

D. P. Bertsekas : *Dynamic Programming and Optimal Control*. Athena Scientific, 2 édition, 2000. ISBN 1886529094.

D. P. Bertsekas et S. E. Shreve : *Stochastic Optimal Control : the discrete-time case*. Athena Scientific, Belmont, 1996.

D. P. Bertsekas et J. N. Tsitsiklis : *Neuro-Dynamic Programming*. Athena Scientific, 1996.

D. P. Bertsekas et J. N. Tsitsiklis : Gradient convergence in gradient methods. *SIAM Journal on Optimization*, 10(3):627–642, 2000.

P. Billingsley : *Probability and Measure*. Wiley-Interscience, 3 édition, 1995.

J. R. Birge et F. V. Louveaux : *Introduction to Stochastic Programming*. Springer Series in Operations Research. Springer Verlag, New York, 1997.

J. F. Bonnans, J.-Ch. Gilbert, C. Lemaréchal et C. Sagastizábal : *Optimisation Numérique*. Springer Verlag, 1997.

B. Bouchard, R. Elie et N. Touzi : Stochastic target problems with controlled loss. *SIAM Journal on Control and Optimization*, 48(5):3123–3150, 2009.

P. Carpentier, J.-Ph. Chancelier, G. Cohen, M. De Lara et P. Girardeau : Dynamic consistency for stochastic optimal control problems. URL http://arxiv.org/pdf/1005.3605v1. arXiv :1005.3605, 5 2010.

P. Carpentier, J.-Ph. Chancelier et M. De Lara : Approximations of stochastic optimization problems subject to measurability constraints. *SIAM Journal on Optimization*, 19(4):1719–1734, 2009a.

P. Carpentier, C. Cohen, J.-C. Culioli et A. Renaud : Stochastic optimization of unit commitment : a new decomposition framework. *IEEE Transactions on Power Systems*, 11(2):1067–1073, 5 1996.

P. Carpentier, G. Cohen et J.-C. Culioli : Stochastic optimal control and decomposition-coordination methods. *In* R. Durier et C. Michelot, éditeurs : *Recent Developments in Optimization. Lecture Notes in Economics and Mathematical Systems*, 429, pages 72–103. Springer Verlag, Berlin, 1995.

P. Carpentier, G. Cohen et A. Dallagi : Particle Methods for Stochastic Optimal Control Problems. arXiv :0907.4663v1, 2009b.

P. Cheridito, F. Delbaen et M. Kupper : Dynamic monetary risk measures for bounded discrete-time processes. *Electronic Journal of Probability*, 11(3):57–106, 2006. URL http://www.math.washington.edu/~ejpecp/.

G. Cohen : Optimization by Decomposition and Coordination : A Unified Approach. *IEEE Transactions on Automatic Control*, 23:222–232, 1978.

G. Cohen : *Décomposition et Coordination en optimisation déterministe différentiable et non-différentiable*. Thèse de doctorat d'État, Université de Paris IX Dauphine, 1984.

G. Cohen : Convexité et Optimisation. Cours de l'École Nationale des Ponts et Chausseés, 2000. URL http://www-rocq.inria.fr/metalau/cohen/documents/Ponts-cours-A4-NB.pdf.

G. Cohen : *Optimisation des Grands Systèmes*. Cours du DEA MMME, Université de Paris I, 2004.

G. Cohen et J.-C. Culioli : Decomposition Coordination Algorithms for Stochastic Optimization. *SIAM J. Control Optimization*, 28(6):1372–1403, 1990.

G. Cohen et D. L. Zhu : Decomposition coordination methods in large scale optimization problems. the nondifferentiable case and the use of augmented Lagrangians. *In* J.B. Cruz, éditeur : *Advances in Large Scale Systems*, 1, pages 203–266. JAI Press Inc., Greenwich, Connecticut, 1984.

J.-C. Culioli : *Algorithmes de décomposition/coordination en optimisation stochastique*. Thèse de doctorat, École des Mines de Paris, 1987.

A. Dallagi : *Méthodes particulaires en commande optimale stochastique*. Thèse de doctorat, Université Paris 1, Panthéon-Sorbonne, 2007.

G. B. Dantzig : Linear programming under uncertainty. *Management Science*, 1:197–206, 1955.

D. P. de Farias et B. Van Roy : The Linear Programming Approach to Approximate Dynamic Programming. *Oper. Res.*, 51(6):850–856, 2003.

F. Delebecque et J.-P. Quadrat : Contribution of Stochastic control Singular Perturbation Averaging and Team Theories to an Example of Large-Scale Systems : Management of Hydropower Production. *IEEE Transactions on Automatic Control*, 23:209–222, 1978.

C. Dellacherie et P. A. Meyer : *Probabilités et potentiel*. Hermann, Paris, 1975.

K. Detlefsen et G. Scandolo : Conditional and dynamic convex risk measures. *Finance and Stochastics*, 9(4):539–561, 10 2005.

L. Devroye : *A course in density estimation*. Birkhauser Boston Inc., Cambridge, MA, USA, 1987. ISBN 0-8176-3365-0.

S. Dreyfus : Richard Bellman on the birth of Dynamic Programming. *Operations Research*, 50(1):49–51, 1-2 2002.

J. Dupačová, N. Gröwe-Kuska et W. Römisch : Scenario reduction in stochastic programming. An approach using probability metrics. *Mathematical Programming*, 95 (3):493–511, 2003.

I. Ekeland et R. Temam : *Convex Analysis and Variational Problems*, volume 28 de *Classics in Applied Mathematics*. SIAM, 1999.

G. Emiel et C. Sagastizábal : Incremental-like Bundle Methods with Application to Energy Planning. *Computational Optimization and Applications*, 46(2):305–332, 2010.

S. Gal : The parameter iteration method in dynamic programming. *Management Science*, 35(6):675–684, 6 1989.

J.-Ch. Gilbert : Éléments d'Optimisation Différentiable. Cours de l'ENSTA-ParisTech, 2010. URL http://www-rocq.inria.fr/~gilbert/ensta/optim.html.

P. Girardeau : A comparison of sample-based Stochastic Optimal Control methods. Stochastic Programming E-Print Series, http://www.speps.org, 2010. submitted to Optimization and Engineering.

R. Glon, K. Barty et P. Girardeau : Méthode d'agrégation appliquée à un problème de gestion de production. Rapport de stage, EDF, 2010.

P. J. Hammond : Consistent plans, consequentialism, and expected utility. *Econometrica*, 57(6):1445–1449, 1989.

T. J. Hastie et R. J. Tibshirani : *Generalized Additive Models*. Chapman & Hall/CRC, 1990.

H. Heitsch et W. Römisch : Scenario Reduction Algorithms in Stochastic Programming. *Computational Optimization and Applications*, 24:187–206, 2003.

H. Heitsch, W. Römisch et C. Strugarek : Stability of multistage stochastic programs. *SIAM Journal on Optimization*, 17:511–525, 2006.

R. Henrion : On the connectedness of probabilistic constraint sets. *Journal of Optimization Theory and Applications*, 112(3):657–663, 2002. ISSN 0022-3239.

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

J. L. Higle et S. Sen : *Stochastic Decomposition : A Statistical Method for Large Scale Stochastic Linear Programming*. Kluwer Academic Publishers, Dordrecht, 1996.

J.-B. Hiriart-Urruty : Extension of lipschitz integrands and minimization of nonconvex integral functionals : Applications to the optimal recourse problem in dicrete time. *Probability and mathematical statistics*, 3(1):19–36, 1982.

B. Kharrat, K. Barty et P. Girardeau : Étude expérimentale d'un algorithme de décomposition stochastique. Rapport de stage H-R36-2009-01815-FR, EDF, 2009.

M. K. Kreps et E. L. Porteus : Temporal resolution of uncertainty and dynamic choice theory. *Econometrica*, 46:185–200, 1978.

P. Lahirigoyen, K. Barty et P. Girardeau : Entre programmation dynamique et décomposition par les prix dans le cadre stochastique : une méthode hybride. Rapport de stage H-R35-2008-03799-FR, EDF, 2008.

P. Lederer, Ph. Torrion et J.-P. Bouttes : Un feedback global pour la planification du parc de production électrique français. *Analysis and Optimization of Systems*, 62: 102–115, 1984.

F. A. Longstaff et E. S. Schwartz : Valuing american options by simulation : A simple least squares approach. *Review of Financial Studies*, 14(1):113–147, 2001.

L. Mori : Writing a thesis with LaTeX. *The PracTeXJournal*, 1, 2008. URL http://tug.org/pracjourn/2008-1/mori/.

E. A. Nadaraya : On estimating regression. *Theory of Probability and its applications*, 10:186–190, 1964.

T. Pennanen : Epi-convergent discretizations of multistage stochastic programs. *Mathematics of Operations Research*, 30:245–256, 2005.

M. V. F. Pereira et L. M. V. G. Pinto : Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(2):359–375, 1991. ISSN 0025-5610.

G. Ch. Pflug : Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming*, 89:251–271, 2001.

A. B. Philpott et Z. Guan : On the convergence of stochastic dual dynamic programming and related methods. *Operations Reseach Letters*, 36(4):450–455, 2008.

W. B. Powell : *Approximate Dynamic Programming*. Wiley Series in Probability and Statistics. John Wiley & Sons, 2007.

A. Prékopa : *Stochastic Programming*. Kluwer, Dordrecht, 1995.

M.L. Puterman : *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. John Wiley and Sons, New York, NY, 1994.

J.-P. Quadrat : Notes de cours de commande optimale stochastique, 2007. URL http://www.librecours.org/documents/96/9681.pdf. Cours du DEA MMME, Université Paris 1.

J.-P. Quadrat, M. Gousat, A. Hertz et M. Viot : Méthodes de gradient stochastique pour l'optimisation des investissements dans un réseau électrique. *In Bulletin de la Direction des Études et Recherches*, pages 1933–1967. EDF, 1981.

J.-P. Quadrat et M. Viot : Introduction à la commande stochastique, 9 1999. URL http://www-rocq.inria.fr/metalau/quadrat/ComSto0.9.pdf. Version préliminaire d'un livre à partir des deux polycopiés de l'École Polytechnique.

R Development Core Team : *R : A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. URL http://www.R-project.org. ISBN 3-900051-07-0.

F. Riedel : Dynamic coherent risk measures. *Stochastic Processes and their Applications*, 112(2):185 – 200, 2004. ISSN 0304-4149. URL http://www.sciencedirect.com/science/article/B6V1B-4C4VXT3-2/2/00948ab87f2c04a301df058f0614439c.

H. Robbins et S. Monro : A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

R. T. Rockafellar et R. J.-B. Wets : *Variational Analysis*. Springer Verlag, Berlin Heidelberg, 1998.

A. Ruszczyński : Risk-averse dynamic programming for markov decision processes. *Optimization Online*, to appear in Mathematical Programming, December 2009. URL http://www.optimization-online.org/DB_HTML/2009/12/2497.html.

A. Ruszczyński et A. Shapiro, éditeurs. *Stochastic Programming*, volume 10 de *Handbooks in Operations Research and Management Science*. Elsevier, 2003.

A. Shapiro : On complexity of multistage stochastic programs. *Operations Research Letters*, 34:1–8, 2006.

A. Shapiro : On a time consistency concept in risk averse multistage stochastic programming. *Operations Research Letters*, 37(3):143 – 147, 2009. ISSN 0167-6377. URL http://www.sciencedirect.com/science/article/B6V8M-4VSB147-3/2/e5c0b0ede2763d1e7d811a439b64736d.

A. Shapiro : Analysis of Stochastic Dual Dynamic Programming Method. *European Journal of Operational Research*, 2010. à paraître.

A. Shapiro, D. Dentcheva et A. Ruszczyński : *Lectures on Stochastic Programming.* Society for Industrial and Applied Mathematics, Philadelphia, 2009.

R. H. Strotz : Myopia and Inconsistency in Dynamic Utility Maximization. *The Review of Economic Studies*, 23(3):165–180, 1955.

C. Strugarek : *Approches variationnelles et autres contributions en optimisation stochastique.* Thèse de doctorat, École Nationale des Ponts et Chaussées, 5 2006.

Ph. Torrion et J. Leveugle : Comparaison de différentes méthodes d'optimisation appliquées à la gestion annuelle du système offre-demande d'électrivité français. Rapport technique, Électricité de France, 1985.

J. Tsitsiklis et B. Van Roy : Optimal stopping for markov processes : Hilbert space theory, approximation algorithm and an application to pricing high-dimensional financial derivatives. *IEEE Transactions on Automatic Control*, 44:1840–1851, 1999.

A. Turgeon : Optimal operation of multi-reservoir power systems with stochastic inflows. *Water Resources Research*, 16(2):275–283, 1980.

P. Vezolle, S. Vialle et X. Warin : Large Scale Experiment and Optimization of a Distributed Stochastic Control Algorithm. Application to Energy Management Problems. *In International workshop on Large-Scale Parallel Processing (LSPP 2009)*, Rome, Italy, 2009. ISBN 978-1-4244-3750-4.

G. S. Watson : Smooth regression analysis. *Shankya Series A*, 26:359–372, 1964.

P. Whittle : *Optimization over time.* John Wiley & Sons, 1982.

H. S. Witsenhausen : A counterexample in stochastic optimal control. *SIAM Journal of Control*, 2(6):149–160, 1968.

H. S. Witsenhausen : A standard form for sequential stochastic control. *Mathematical Systems Theory*, 7(1):5–11, 1973.

S. N. Wood : *Generalized Additive Models : An Introduction with R.* Chapman & Hall/CRC, 2006.