



# **Optimisation stochastique des systèmes multi-réservoirs par l'agrégation de scénarios et la programmation dynamique approximative**

**Thèse**

**Luckny Zéphyr**

**Doctorat en sciences de l'administration – Opérations et systèmes de  
décision**

Philosophiæ doctor (Ph.D.)

Québec, Canada

© Luckny Zéphyr, 2015



# Résumé

Les problèmes de gestion des réservoirs sont stochastiques principalement à cause de l'incertitude sur les apports naturels. Ceci entraîne des modèles d'optimisation de grande taille pouvant être difficilement traitables numériquement.

La première partie de cette thèse réexamine la méthode d'agrégation de scénarios proposée par Rockafellar et Wets (1991). L'objectif consiste à améliorer la vitesse de convergence de l'*algorithme du progressive hedging* sur lequel repose la méthode.

L'approche traditionnelle consiste à utiliser une valeur fixe pour ce paramètre ou à l'ajuster selon une trajectoire choisie a priori : croissante ou décroissante. Une approche dynamique est proposée pour mettre à jour le paramètre en fonction d'information sur la convergence globale fournie par les solutions à chaque itération. Il s'agit donc d'une approche a posteriori.

La thèse aborde aussi la gestion des réservoirs par la programmation dynamique stochastique. Celle-ci se prête bien à ces problèmes de gestion à cause de la nature séquentielle de leurs décisions opérationnelles. Cependant, les applications sont limitées à un nombre restreint de réservoirs. La complexité du problème peut augmenter exponentiellement avec le nombre de variables d'état, particulièrement quand l'approche classique est utilisée, i.e. en discrétisant l'espace des états de « manière uniforme ».

La thèse propose une approche d'approximation sur une grille irrégulière basée sur une décomposition simpliciale de l'espace des états. La fonction de valeur est évaluée aux sommets de ces simplexes et interpolée ailleurs. À l'aide de bornes sur la vraie fonction, la grille est raffinée tout en contrôlant l'erreur d'approximation commise.

En outre, dans un contexte décision-information spécifique, une hypothèse « uni-bassin », souvent utilisée par les hydrologues, est exploitée pour développer des formes analytiques pour l'espérance de la fonction de valeur.

Bien que la méthode proposée ne résolve pas le problème de complexité non polynomiale de la programmation dynamique, les résultats d'une étude de cas industrielle montrent qu'il n'est pas forcément nécessaire d'utiliser une grille très dense pour approximer la fonction de valeur avec une précision acceptable. Une bonne approximation pourrait être obtenue en évaluant cette fonction uniquement en quelques points de grille choisis adéquatement.

# Abstract

Reservoir operation problems are in essence stochastic because of the uncertain nature of natural inflows. This leads to very large optimization models that may be difficult to handle numerically.

The first part of this thesis revisits the scenario aggregation method proposed by Rochafellar and Wets (1991). Our objective is to improve the convergence of the progressive hedging algorithm on which the method is based. This algorithm is based on an augmented Lagrangian with a penalty parameter that plays an important role in its convergence.

The classical approach consists in using a fixed value for the parameter or in adjusting it according a trajectory chosen a priori: decreasing or increasing. This thesis presents a dynamic approach to update the parameter based on information on the global convergence provided by the solutions at each iteration. Therefore, it is an a posteriori scheme.

The thesis also addresses reservoir problems via stochastic dynamic programming. This scheme is widely used for such problems because of the sequential nature of the operational decisions of reservoir management. However, dynamic programming is limited to a small number of reservoirs. The complexity may increase exponentially with the dimension of the state variables, especially when the classical approach is used, i.e. by discretizing the state space into a "regular grid".

This thesis proposes an approximation scheme over an irregular grid based on simplicial decomposition of the state space. The value function is evaluated over the vertices of these simplices and interpolated elsewhere. Using bounds on the true function, the grid is refined while controlling the approximation error.

Furthermore, in a specific information-decision context, a "uni-bassin" assumption often used by hydrologists is exploited to develop analytical forms for the expectation of the

value function.

Though the proposed method does not eliminate the non-polynomial complexity of dynamic programming, the results of an industrial case study show that it is not absolutely necessary to use a very dense grid to appropriately approximate the value function. Good approximation may be obtained by evaluating this function at few appropriately selected grid points.

# Table des matières

Résumé	iii
Abstract	v
Table des matières	vii
Liste des tableaux	xi
Liste des figures	xiii
Liste des abréviations	xv
Remerciements	xix
<b>1 Introduction générale</b>	<b>1</b>
1.1 Opérations des systèmes de réservoirs . . . . .	2
1.2 Représentation d'un système multi-réservoirs . . . . .	3
1.3 Approches numériques pour la gestion des systèmes hydriques . . . . .	5
1.3.1 Optimisation stochastique implicite . . . . .	6
1.3.2 Optimisation stochastique explicite . . . . .	9
1.3.3 Méthodes heuristiques . . . . .	11
1.4 Objet et problématique de la thèse . . . . .	12
1.5 Contributions et organisation de la thèse . . . . .	13
<b>2 Méthode d'agrégation de scénarios</b>	<b>15</b>
2.1 Introduction . . . . .	15
2.2 Formulation du programme mathématique . . . . .	16
2.3 Approches de décomposition . . . . .	19
2.4 Décomposition par scénarios et l'algorithme du <i>progressive hedging</i> . . . . .	20
2.5 Ajustement du paramètre de pénalisation . . . . .	22
2.5.1 Une heuristique d'apprentissage pour ajuster le paramètre . . . . .	24
2.5.2 Initialisation du paramètre de pénalisation . . . . .	27
2.6 Expériences numériques . . . . .	29
2.6.1 Ensemble test . . . . .	29
2.6.2 Analyses et résultats . . . . .	31
2.7 Conclusion . . . . .	33

<b>3</b>	<b>Méthode d'approximation simpliciale dans le cadre de la programmation dynamique</b>	<b>35</b>
3.1	Introduction . . . . .	35
3.2	Formulation du programme mathématique . . . . .	36
3.3	Partition de l'espace des stocks en simplexes . . . . .	37
3.3.1	Représentations d'un simplexe . . . . .	38
3.3.2	Faces d'un simplexe . . . . .	39
3.3.3	Division d'un pavé en simplexes . . . . .	39
3.3.4	Division d'un simplexe . . . . .	44
3.4	Approximation de la fonction de valeur . . . . .	47
3.5	Interpolation de la fonction de valeur . . . . .	50
3.6	Expériences numériques . . . . .	53
3.6.1	Cadre expérimental . . . . .	53
3.6.2	Résultats . . . . .	55
3.6.3	Remarques empiriques . . . . .	56
3.7	Conclusion . . . . .	57
<b>4</b>	<b>Méthode de programmation dynamique approximative</b>	<b>59</b>
4.1	Introduction . . . . .	59
4.2	Structure et contrôle d'un système de réservoirs . . . . .	60
4.3	Programmation dynamique approximative . . . . .	65
4.4	Calcul de l'espérance de la fonction de valeur . . . . .	67
4.4.1	Forme analytique . . . . .	69
4.4.2	Cas de la distribution normale tronquée . . . . .	70
4.4.3	Cas de la distribution log-normale . . . . .	70
4.5	Optimisation par la programmation linéaire généralisée . . . . .	71
4.6	Expériences numériques . . . . .	75
4.6.1	Ensemble test . . . . .	76
4.6.2	Résultats et analyses . . . . .	79
4.7	Conclusion . . . . .	81
<b>5</b>	<b>Étude de cas : Rio Tinto Alcan</b>	<b>83</b>
5.1	Introduction . . . . .	83
5.2	Description du système à l'étude . . . . .	84
5.3	Formulation linéaire du programme dynamique . . . . .	90
5.3.1	Discrétisation des débits turbinés . . . . .	92
5.4	Calcul de l'espérance de la fonction $\hat{G}_t$ . . . . .	98
5.5	Cadre expérimental . . . . .	101
5.5.1	Résultats et analyses . . . . .	102
5.5.1.1	Modèle sans variable hydrologique . . . . .	103
5.5.1.2	Modèle avec variable hydrologique . . . . .	105
5.6	Conclusion . . . . .	110
	<b>Conclusion</b>	<b>113</b>







# Liste des tableaux

1.1	Illustration de la complexité de la PD . . . . .	8
2.1	Algorithme du <i>progressive hedging</i> . . . . .	23
2.2	Heuristique d'apprentissage pour $\rho$ . . . . .	25
2.3	Plan de simulation . . . . .	29
2.4	Procédure de construction d'un arbre de scénarios . . . . .	30
2.5	Réalisations possibles des apports naturels et de la demande . . . . .	31
2.6	Statistiques descriptives . . . . .	32
2.7	Tests de signification de moyennes . . . . .	32
2.8	Comparaison de performance : problèmes originaux et perturbés . . . . .	33
3.1	Faces d'un simplexe de dimension $p$ . . . . .	39
3.2	Exemple de nombre de faces de $P$ . . . . .	41
3.3	Comparaison de méthodes de découpage de pavé . . . . .	43
3.4	Temps de calcul en secondes : rehaussement et CFK . . . . .	44
3.5	Algorithme de division de simplexes . . . . .	47
3.6	Algorithme de localisation d'un point . . . . .	51
3.7	Algorithme de traversée de simplexes dans une direction donnée . . . . .	52
3.8	Bornes numériques sur les paramètres des modèles . . . . .	54
3.9	Nombre d'exemples de problèmes . . . . .	54
3.10	Caractéristiques des problèmes . . . . .	55
4.1	Bornes numériques sur les paramètres du modèle à 5 réservoirs . . . . .	77
4.2	Nombre de duplications par valeur de $\eta$ . . . . .	77
4.3	Statistiques sur le temps d'exécution (en secondes) des simulations . . . . .	79
4.4	Statistiques sur le nombre moyen de points de grille par période . . . . .	80
5.1	Statistiques sur l'erreur d'interpolation linéaire pour chaque stratégie de discrétisation . . . . .	97
5.2	Temps de résolution en secondes du problème pour chaque stratégie de discrétisation des débits . . . . .	98
5.3	Résultats de l'analyse de régression . . . . .	101
5.4	Comparaison entre le modèle RTA, MSVH et le modèle $MSVH^{CFK}$ : efficacité moyenne de la politique de gestion et nombre de points de discrétisation des stocks par semaine . . . . .	104

5.5	Compromis entre l'efficacité globale moyenne et le temps de calcul pour les modèles sans variable hydrologique . . . . .	105
5.6	Comparaison entre les modèles MAVH et MSVH : efficacité moyenne de la politique de gestion, nombre moyen de points de discrétisation des stocks par semaine et temps de résolution . . . . .	107

# Liste des figures

1.1	Illustration de la hauteur de chute dans une centrale . . . . .	2
1.2	Exemple d'un système multi-réservoirs . . . . .	4
2.1	Exemple d'un arbre d'événements . . . . .	17
2.2	Scénarios de l'arbre d'événements de la figure 2.1 . . . . .	20
2.3	L'indicateur de progrès $\sigma^\nu$ en fonction de $\tau^\nu$ . . . . .	26
2.4	L'indicateur d'équilibre stabilisé $h^\nu$ en fonction de $\alpha^\nu$ . . . . .	26
2.5	Configuration des réservoirs . . . . .	30
3.1	Décomposition CFK d'un cube . . . . .	40
3.2	Exemples de rehaussement . . . . .	43
3.3	Illustration de l'écart d'approximation $\Delta$ . . . . .	45
3.4	Illustration de la division de simplexes . . . . .	46
3.5	Exemple de partitionnement de l'espace des stocks en simplexes . . . . .	48
3.6	Exemple de traversée de simplexes . . . . .	52
3.7	Configurations de réservoirs . . . . .	53
3.8	Exemple de grille pour un modèle à deux réservoirs . . . . .	55
3.9	Moyennes de l'écart d'approximation et intervalles de confiance à 95% sur les itérations . . . . .	56
3.10	Exemples de cas où les simplexes sont trop plats . . . . .	57
4.1	Exemple de trajectoire des stocks finaux . . . . .	69
4.2	Exemple d'une fonction de production et son approximation . . . . .	72
4.3	Configuration à 5 réservoirs utilisé dans les simulations . . . . .	77
4.4	Deux exemples de fonctions de production utilisées dans les simulations . . . . .	78
4.5	Temps moyen de résolution des problèmes en fonction de l'imprécision dans un repère semi-log . . . . .	80
5.1	Bassin hydrographique du système de RTA . . . . .	84
5.2	Représentation du sous-système étudié . . . . .	85
5.3	Fonctions de production des centrales CCD, CCS et CIM en fonction des débits . . . . .	86
5.4	Fonction de production de la centrale SH . . . . .	86
5.5	Débit maximal en $m^3/s$ des centrales CD et IM en fonction du stock initial . . . . .	87
5.6	Délestage maximal en $m^3/s$ par l'issue PDECH en fonction du volume d'eau stocké au LSJ . . . . .	88

5.7	Fonction de perte de puissance en $MW$ . . . . .	88
5.8	Limites du réservoir LSJ . . . . .	89
5.9	Illustration des points de discrétisation candidats des fonctions de production de la centrale CD . . . . .	93
5.10	Illustration des points de discrétisation candidats des fonctions de production de la centrale CS . . . . .	94
5.11	Illustration des points de discrétisation candidats des fonctions de production de la centrale IM . . . . .	94
5.12	Illustration des points de discrétisation candidats des fonctions de production de la centrale SH . . . . .	95
5.13	Illustration de la discrétisation $ST^1$ . . . . .	96
5.14	Apports naturels hebdomadaires totaux historiques . . . . .	99
5.15	Trois exemples de grille à trois périodes différentes et un exemple de grille régulière . . . . .	105
5.16	Trajectoire hebdomadaire optimale des réservoirs prescrite par le modèle RTA sur les 25 années de simulation . . . . .	108
5.17	Trajectoire hebdomadaire optimale des réservoirs prescrite par le MSVH sur les 25 années de simulation . . . . .	109
5.18	Trajectoire optimale moyennes des réservoirs prescrite par les modèles RTA et MSVH . . . . .	109
5.19	Évolution du temps de calcul en fonction du nombre de valeurs de la variable hydrologique . . . . .	110

# Liste des abréviations

$hm^3$	hectomètres cube
$m^3/s$	mètres cube par seconde
AG	Algorithme génétique
APH	Algorithme du <i>Progressive Hedging</i>
CCD	Centrale Chute-Du-Diable
CCS	Centrale Chute-Savanne
CD	Chute-Du-Diable
CIM	Centrale Isle-Maligne
CSH	Centrale Shipshaw
IM	Isle-Maligne
LSJ	Lac Saint-Jean
MAVH	Modèle avec variable hydrologique
MSVH	Modèle sans variable hydrologique
MW	Mega watts
NA	Non anticipativité
PD	Programmation dynamique
PDS	Programmation dynamique stochastique
PDSD	Programmation dynamique stochastique duale
PL	Programmation linéaire
PLG	Programmation linéaire généralisée

PND Programmation neuro-dynamique  
RN Réseaux de neurones  
RTA Rio Tinto Alcan  
SH Shipshaw  
SSDP *Sampling Stochastic Dynamic Programming*



*À la mémoire de mon père*



# Remerciements

Je remercie mon directeur, Bernard F. Lamond, et mes codirecteurs de thèse, Pascal Lang et Pascal Côté, qui ont supervisé mes travaux de recherche. Bernard F. Lamond m'a introduit aux problèmes de réservoirs. Pascal Lang m'a aidé à faire mes premiers pas en analyse convexe. Pascal côté m'a ouvert les portes pour l'application d'une partie des travaux au contexte industriel. C'était une expérience enrichissante de travailler avec cette équipe.

De chaleureux remerciements aux professeurs Adnène Hajji, du département Opérations et système de décision de l'Université Laval, Amaury Tilmant, du département de génie civil et génie des eaux de l'Université Laval, et Matt Davison du département des mathématiques appliquées de l'Université Western Ontario qui ont accepté de faire partie de mon jury de thèse.

Mes remerciements vont aussi aux professeurs du département Opérations et systèmes de décision de la Faculté des sciences de l'administration de l'Université Laval qui m'ont tenu la main durant ma scolarité de doctorat. Je pense particulièrement au professeur David Beaudoin qui, à l'aide de ses fonds de recherche, a financé une partie de ma bourse pendant plus de deux ans.

J'ai une pensée spéciale pour ma famille, ma mère, mes frères et sœurs, mes beaux-frères et ma belle-sœur, mes neveux et nièces qui m'ont encouragé tout au long de cette aventure. Faire un doctorat peut être une longue aventure !

En plus de ma mère, je dois des remerciements particuliers à mon grand frère Lemète Zéphyr et ma grande sœur Bénila Zéphyr, qui, en plus de m'avoir toujours motivé à faire des études supérieures, ont contribué financièrement à mon éducation depuis l'école primaire jusqu'à mes études de premier cycle universitaire comme si j'étais leur fils.

Les encouragements de ma famille m'ont été très utiles pour terminer ce doctorat.

Je remercie également mes amies et amis, et mes collègues du département Opérations et systèmes de décision pour leurs encouragements. Je me garde de citer de nom afin de n'oublier personne.

Je dois remercier aussi l'entreprise Rio Tinto Alcan d'avoir accepté que je teste la méthode développée dans la deuxième partie de la thèse à son système. La porte de l'entreprise m'a toujours été ouverte pour des visites des installations et un stage à la division Énergie électrique au Saguenay.

Je tiens aussi à mentionner la contribution financière du Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) et de Rio Tinto Alcan dans mes travaux de recherche.

En rédigeant ces propos, je pense à mon grand frère Daniel Zéphyr qui vit avec un retard mental grave. Bientôt âgé de 37 ans, je ne sais pas si, dans sa tête, Daniel est un enfant, un adolescent, un jeune ou un adulte. La seule chose dont je suis certain est que je ne pourrai jamais lui expliquer les raisons qui m'ont motivé à faire cette thèse ni le problème d'incertitude auquel celle-ci s'intéresse. Ça aussi fait partie des incertitudes de la vie!

À défaut d'écrire un livre sur les préoccupations que sa réalité de vie suscitent en moi, je dédie cette thèse à Daniel avec une immense affection.

# Chapitre 1

## Introduction générale

L'eau est une source de production d'énergie électrique qui offre de nombreux avantages. C'est une ressource renouvelable et, contrairement à d'autres centrales (par exemple les centrales thermiques), les centrales hydroélectriques n'émettent pas de gaz polluants qui causent le smog et des pluies acides (Vandal, 2012). Pour profiter de l'énergie potentielle de l'eau, des ouvrages de retenue (réservoirs) sont construits sur les lacs ou rivières à des fins de stockage et de contrôle de la hauteur de chute dans les centrales.

Pour produire l'hydroélectricité, les ressources hydriques sont exploitées dans un contexte d'incertitude, principalement sur les apports naturels aux réservoirs. En effet, on ne peut pas prévoir les précipitations futures avec certitude. D'où se pose un problème de planification de la gestion efficace de l'eau dans un environnement incertain. Par exemple, il faut utiliser l'eau pour produire de l'énergie pour répondre aux besoins présents sans compromettre la production future. Il est donc nécessaire de déterminer des plans de turbinage et de stockage basés, entre autres, sur l'information disponible sur le niveau des réservoirs et des prévisions sur les apports naturels futurs.

Depuis des décennies, des modèles mathématiques ont été étudiés pour déterminer des « politiques » de gestion optimale des systèmes de réservoirs. On attribue à Little (1955) le développement du premier modèle numérique pour la résolution d'un programme d'optimisation d'un système à un seul réservoir. Au début, la recherche dans ce domaine se limitait à des systèmes à un seul réservoir à cause des limites de mémoire et de vitesse des ordinateurs d'alors. Avec l'amélioration de la performance des ordinateurs, quelques années plus tard, un ensemble de méthodes d'optimisation spécialisées ont été proposées afin de résoudre des modèles multi-réservoirs, dont des méthodes de décomposition comme l'agrégation de scénarios et la programmation dynamique.

## 1.1 Opérations des systèmes de réservoirs

Un système hydroélectrique est généralement composé d'un réseau de centrales parfois en aval de réservoirs. Ces réservoirs peuvent être disposés en arborescence suivant la topologie des rivières et de leurs affluents. Une centrale hydroélectrique est une usine où l'énergie potentielle de l'eau est exploitée pour faire tourner des turbines. Une fois celles-ci mises en marche par la force motrice de l'eau, l'énergie potentielle est transformée en énergie mécanique qui, à son tour, actionne des alternateurs qui convertissent cette dernière en énergie électrique.

La quantité d'énergie produite dans chaque centrale est une fonction de la hauteur de chute ( $H_{it}$ ) dans la centrale et du débit turbiné. La hauteur de chute (brute) est la différence entre la hauteur de l'eau dans le réservoir ( $h_{it}$ ) et le niveau de restitution en aval ( $h_{ir}$ ) (figure 1.1, Anctil *et al.*, 2012). Cependant, la hauteur de chute effective dans la turbine est plus faible que cette différence. En fait, elle subit des variations, d'une part, à cause de la variation des niveaux amont et aval, et d'autre part, à cause des frictions dans les conduites d'amenée de l'eau (Hammadia, 2001).

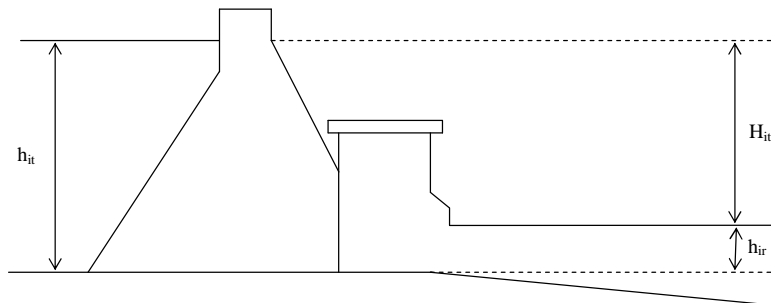


FIGURE 1.1 – Illustration de la hauteur de chute dans une centrale

La modélisation et l'intégration de la hauteur de chute dans un modèle d'optimisation peuvent s'avérer difficiles, particulièrement quand les niveaux de l'eau en amont et en aval dépendent des débits turbinés. Dans le cas de grands réservoirs, les effets de hauteur de chute peuvent être insensibles aux débits turbinés si les centrales sont de faible capacité. Les gestionnaires veulent souvent avoir une hauteur de chute élevée afin de maximiser le rendement des turbines. La relation entre les débits turbinés et la hauteur de chute peut alors exiger une attention particulière (Pérez-Díaz *et al.*, 2010).

Généralement, la gestion des réservoirs est divisée en trois catégories : long, moyen et

court termes. Les modèles de long terme servent à l'évaluation de la capacité de production des centrales en réponse à l'évolution du prix de l'électricité et/ou de la demande, à la modification des ouvrages et à la négociation des contrats d'approvisionnement. Ils peuvent consister aussi à déterminer la capacité optimale des réservoirs étant donné des demandes en eau et des apports naturels aléatoires. Le pas de temps est généralement le mois ou l'année, et l'horizon de planification est souvent de plusieurs années (Côté, 2010).

Les modèles à moyen terme servent généralement à planifier l'opération des réservoirs sur un horizon d'une année ou plus, mais jamais de plus de trois ans (Côté, 2010). L'objectif est de trouver une politique de gestion afin d'optimiser certains critères de performance, comme maximiser la production hydroélectrique ou les revenus d'opération ou encore minimiser les coûts de production. Ces modèles sont caractérisés par des fonctions de production non linéaires et non concaves. Le pas de temps peut être mensuel ou hebdomadaire.

Une fois les politiques de gestion à moyen terme obtenues, celles-ci servent d'entrées à des modèles de court terme pour obtenir les flux d'eau à turbiner en temps réel (par heure ou par jour) sur un horizon de planification de l'ordre de la semaine. Pour chaque pas de temps, l'opérateur du système décide du flux à turbiner dans chaque centrale en fonction du volume d'eau disponible. Ces modèles servent aussi à déterminer la répartition de la production entre les différents groupes turbo-alternateurs. Ils intègrent des fonctions fortement non linéaires et non convexes.

## 1.2 Représentation d'un système multi-réservoirs

Un système multi-réservoirs peut être représenté par un graphe où chaque nœud correspond à un réservoir seul ou une centrale seule (au fil de l'eau, c'est-à-dire sans réservoirs) ou encore un groupe centrale-réservoir. Les nœuds sont numérotés de 1 à  $p$  selon un *tri topologique*. La figure 1.2 montre un exemple avec six nœuds. Les triangles représentent des réservoirs, les rectangles des centrales, les arcs pleins les flux turbinés ou délestés (évacués) en cas de trop-plein et les arcs discontinus des apports naturels non contrôlés.

Ce graphe est constitué de deux réseaux correspondant aux écoulements planifiés (turbinés) et au délestage (déversement du surplus d'eau). La topographie de ces réseaux

est décrite via les matrices d'incidence de dimension  $p \times p$  suivantes :

$$B_{ij} = \begin{cases} 1 & \text{si } i = j \\ -1 & \text{si les écoulements planifiés sortant du nœud } j \text{ sont acheminés au nœud } i \\ 0 & \text{dans tous les autres cas} \end{cases}$$

et

$$C_{ij} = \begin{cases} 1 & \text{si } i = j \\ -1 & \text{si l'eau délestée du nœud } j \text{ peut être acheminée au nœud } i \\ 0 & \text{dans tous les autres cas} \end{cases}$$

Par exemple, dans la figure 1.2 :

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \text{ et } C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

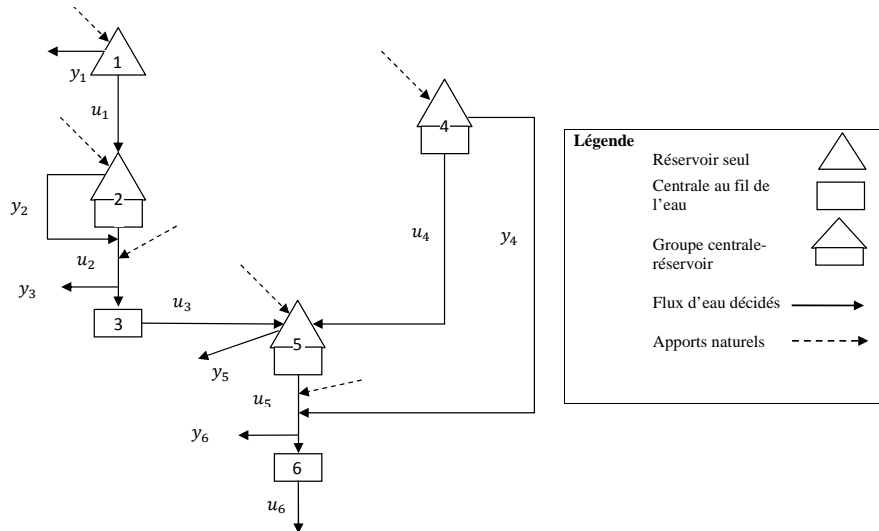


FIGURE 1.2 – Exemple d'un système multi-réservoirs

L'état du système est décrit par les stocks d'eau disponibles dans les réservoirs au début ou à la fin de la période  $t$ , notés  $v_t$ . À chaque période  $t$ , l'opérateur du système doit



prendre deux types de décision : les écoulements planifiés des réservoirs notés  $u_t$  et les évacuations de trop-plein ou délestage notés  $y_t$ .

Étant donné qu'en plus de la production d'énergie électrique, les systèmes de réservoirs sont souvent utilisés à d'autres fins telles que le contrôle d'inondation, l'irrigation, les activités de récréation, des limites sont imposées sur le niveau d'eau dans les réservoirs. À chaque période  $t$ , on note respectivement par  $\underline{v}_t$  et  $\bar{v}_t$  les bornes inférieures et supérieures des réservoirs, et respectivement par  $\underline{u}_t$  et  $\bar{u}_t$  les limites des flux turbinés.

Des modèles d'optimisation sont souvent utilisés afin de déterminer, à chaque période, les décisions qui maximisent la production d'électricité ou qui minimisent les coûts de production. Un modèle d'optimisation général à moyen terme s'écrit par exemple :

$$\text{Max } E \left[ \sum_{t=1}^T f_t(v_{t-1}, u_t, Q_t) + V_{T+1}(v_{T+1}) \right] \quad (1.1)$$

$$\text{S.l.c. } v_t = v_{t-1} - Bu_t - Cy_t + Q_t \quad t = 1, \dots, T \quad (1.2)$$

$$v_t \leq v_t \leq \bar{v}_t \quad t = 1, \dots, T \quad (1.3)$$

$$u_t \leq u_t \leq \bar{u}_t \quad t = 1, \dots, T \quad (1.4)$$

$$y_t \geq 0 \quad t = 1, \dots, T \quad (1.5)$$

où  $E$  dénote l'espérance mathématique,  $Q_t$  est un vecteur de variables aléatoires, en l'occurrence les apports naturels.  $f_t$  représente par exemple une fonction de production ou de revenu.  $V_{T+1}(v_{T+1})$  est la valeur terminale de l'eau.

Des formulations contextuelles sont présentées dans Shawwash *et al.* (2000), Barros *et al.* (2003), Yurtal *et al.* (2005), Kim *et al.* (2007) et dans les chapitres 2, 4 et 5.

### 1.3 Approches numériques pour la gestion des systèmes hydriques

Plusieurs approches numériques ont été proposées pour résoudre le problème (1.1–1.5). En général, les approches d'optimisation sont regroupées en deux grandes catégories : la programmation stochastique implicite, discutée dans la section 1.3.1, et la programmation stochastique explicite, exposée dans la section 1.3.2. Il existe aussi des approches heuristiques dont nous discuterons brièvement dans la section 1.3.3. Les trois chapitres suivants compléteront cette revue par la discussion de certains travaux spécifiques à leur objet d'étude.

La planification à moyen terme des systèmes multi-réservoirs est caractérisée notamment par l'incertitude sur les apports naturels et éventuellement la demande. Il s'agit donc d'un problème stochastique. La principale différence entre les approches d'optimisation implicite et explicite réside principalement dans la façon d'intégrer l'incertitude dans les modèles d'optimisation.

### 1.3.1 Optimisation stochastique implicite

Comme leur appellation l'indique, les méthodes d'optimisation stochastique implicite, ou optimisation de Monte Carlo, intègrent l'incertitude sur les apports hydrauliques de manière implicite dans les modèles d'optimisation (par ex. corrélation sérielle et spatiale). Ces méthodes déterminent des plans de turbinage optimaux à l'aide de longues séries historiques de données hydrologiques (e.g. les apports naturels) ou des générateurs de données hydrologiques synthétiques (Rani et Moreira, 2010). Plusieurs séries (scénarios) d'apports sont générées, et pour chaque scénario, une « politique anticipative » d'opération est déterminée à l'aide d'un modèle d'optimisation déterministe.

Cette procédure d'optimisation stochastique comporte les trois étapes suivantes (Celeste et Billib, 2009) :

- Générer  $M$  séquences de séries hydrologiques de  $N$  périodes chacune.
- À l'aide d'un modèle d'optimisation déterministe, trouver, pour chaque série, les débits turbinés optimaux à chaque période.
- Se servir des  $M \times N$  politiques anticipatives pour construire les règles d'opération futures.

En supposant que la politique d'opération a une structure prédéfinie, les résultats des modèles déterministes sont ensuite utilisés pour déterminer les règles d'opération futures conditionnées à des informations observables comme les niveaux des réservoirs, les apports précédents et/ou les apports prévus (Celeste et Billib, 2009 ; Li *et al.*, 2010).

Si la fonction objectif et les contraintes sont linéaires, la programmation linéaire (PL) pure est souvent utilisée à la deuxième étape de la procédure précédente (e.g. Crawley et Dandy, 1993 ; Reis *et al.*, 2006 ; Needham *et al.*, 2000 ; Shawwash *et al.*, 2000). La PL est souvent utilisée dans les modèles de réservoirs justement à cause de sa capacité à résoudre des modèles de grande taille, la garantie de la convergence vers des solutions optimales globales et la disponibilité de solveurs efficaces à faible coût (Labadie, 2004).

Par exemple, dans l'étude de Reis *et al.* (2006), 36 années d'apports historiques sont

utilisés. Les résultats montrent une relation linéaire entre les stocks d'eau et les flux d'eau turbinés, ce qui pourrait suggérer l'utilisation de règles de décision linéaires pour déterminer les plans de turbinage futurs. Toutefois, ces règles de décision ne sont pas construites dans l'étude.

Les fonctions de production sont généralement non linéaires et non convexes. Pour pouvoir utiliser la PL, des techniques de régularisation comme la PL successive (Mousavi et Ramamurthy, 2000; Barros *et al.*, 2003), où les fonctions sont linéarisées dans le voisinage d'une solution par une approximation d'ordre un, et des extensions comme la PL en nombres entiers mixtes (Chang *et al.*, 2001; Tu *et al.*, 2003) ont été proposées. Néanmoins, ces méthodes sont moins efficaces du point de vue numérique, ce qui peut limiter leur intérêt (Labadie, 2004). En effet, les problèmes en nombres entiers mixtes sont non convexes. Dans le cas de la PL successive, la convergence est souvent lente si la solution optimale n'est pas un point extrême du domaine (linéarisé) (Bazaraa *et al.*, 2013). De plus, il peut être difficile d'adapter la PL successive dans le cas stochastique.

Les problèmes de réservoirs sont souvent résolus sur un horizon de planification fini  $T$ . Pour chaque série hydrologique, il revient donc à trouver une politique d'opération à chaque période de planification  $t$ . La programmation dynamique (PD) déterministe est souvent utilisée dans ce contexte parce qu'elle permet d'exploiter la structure séquentielle du modèle mathématique complexe en le décomposant en  $T$  sous-programmes plus petits et résolus de façon coordonnée. Puisque le pas de temps est discret, on parle de PD en temps discret. De plus, la PD peut facilement intégrer des fonctions non linéaires et non convexes (les chapitres 3, 4 et 5 donnent des exemples).<sup>1</sup>

Partant du principe d'optimalité de Bellman, le modèle mathématique est résolu en procédant par *induction à rebours* pour  $t = T, T - 1, \dots, 1$  en évaluant, pour chaque état possible du système  $v_t$ , la *fonction de valeur* suivante :

$$V_t(v_t) = \max_{u_t, y_t} \{f_t(u_t, v_t) + V_{t+1}(v_{t+1})\}$$

sous un ensemble de contraintes opérationnelles.  $V_{t+1}$  est la fonction de valeur de l'eau à la période  $t + 1$ .

Les premières applications de la PD dans les modèles de réservoirs remontent aux travaux de Little (1955), Young (1967) et Hall *et al.* (1968). Yakowitz (1982) présente une revue des diverses applications de la PD à la gestion des ressources hydriques.

---

1. La PD est basée sur le principe d'optimalité de Bellman (Bellman, 1957) qui stipule : "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decision must constitute an optimal policy with regard to the state resulting from the first decision."

D'autres exemples plus récents sont présentés dans Karamouz et Houck (1982) et Pérez-Díaz *et al.* (2010). Dans Karamouz et Houck (1982), 1000 années d'apports naturels synthétiques sont utilisés. Les plans de turbinage futurs sont déterminés par un modèle de régression multiple ; les flux turbinés à chaque période sont supposés une fonction du volume stocké et des apports naturels au cours de la même période. Toutefois, aucun résultat sur la qualité des régressions (coefficient de détermination, signification des coefficients) n'est fourni.

Depuis son introduction dans la littérature, la PD a été limitée à des modèles à deux ou trois réservoirs à cause du problème de *malédiction de la dimensionnalité*. Par exemple, si le système est composé de  $n$  réservoirs et si les stocks sont discrétisés en  $k$  niveaux chacun, le nombre d'états est  $k^n$ . La table 1.1 illustre cette complexité.

TABLE 1.1 – Illustration de la complexité de la PD

n	k	# d'états
2	7	49
5	7	16 807
7	7	823 543
10	7	282 475 249
13	7	96 889 010 407
15	7	4 747 561 509 943

Pour mitiger ces limites, plusieurs modifications de la PD ont été proposées, dont les techniques d'interpolation de la fonction de valeur (Johnson *et al.*, 1993), la PD avec approximations successives (Shim *et al.*, 2002 ; Yi *et al.*, 2003), où le programme mathématique est décomposé en une suite de programmes unidimensionnels en optimisant sur une variable d'état à la fois et maintenant les autres variables à leur valeur courante, et la PD incrémentée (Yurtal *et al.*, 2005) qui réduit, à chaque itération, l'espace des états en un corridor (sous-domaine) autour de la nouvelle solution.

Les désavantages de la PD incrémentée, dont la sensibilité à des solutions initiales, sont discutés dans Labadie (2004). Dans l'étude de Yurtal *et al.* (2005), 1000 années d'apports synthétiques sont générés ; la valeur moyenne ainsi qu'un intervalle de confiance sur la production sont calculés à partir des résultats obtenus avec ces différentes séries. Aucune règle de construction des plans de turbinage futur n'est mentionnée dans l'article.

### 1.3.2 Optimisation stochastique explicite

Les règles d'opération obtenues avec l'approche implicite ne sont en général pas optimales. En effet, elles sont inférées en se basant sur des plans optimaux ajustés de façon indépendante à partir de scénarios considérés adéquats pour représenter l'incertitude sur les apports naturels.

À l'aide de plusieurs exemples hypothétiques, Philbrick et Kitanidis (1999) montrent les limites des méthodes d'optimisation déterministe. Celles-ci ne sont pas appropriées pour des conditions d'opération extrêmes telles que les situations de sécheresse sévère ou d'inondation, et peuvent produire des solutions de mauvaise qualité, même dans des conditions normales, car elles ne permettent pas toujours de se prémunir adéquatement contre des imprévus.

Contrairement à l'approche implicite, l'optimisation stochastique explicite tient compte de l'incertitude sur les apports naturels en intégrant leur distribution de probabilité directement dans le modèle d'optimisation.

Par exemple, supposant que les apports naturels sont des processus stochastiques sériellement indépendants, la programmation dynamique stochastique (PDS) cherche une solution optimale au programme mathématique suivant :

$$V_t(v_t) = \max_{u_t, y_t} \{E_{Q_t} [f_t(u_t, v_t, Q_t) + V_{t+1}(v_{t+1})]\}$$

sous des contraintes opérationnelles.

Si, au contraire, on suppose que les apports sont conditionnés à une variable hydrologique  $\tilde{\epsilon}_{t-1}$ , le programme devient :

$$V_t(v_t, \epsilon_{t-1}) = \max_{u_t, y_t} \left\{ E_{Q_t | \epsilon_{t-1}} \left[ f_t(u_t, v_t, Q_t) + E_{\tilde{\epsilon}_{t+1} | \epsilon_t} [V_{t+1}(v_{t+1}, \epsilon_t)] \right] \right\}$$

sous les mêmes contraintes.

Il existe plusieurs variantes de la PDS dont le *Sampling Stochastic Dynamic Programming* (SSDP) développé par Kelman *et al.* (1990). Dans cette approche, l'incertitude est intégrée dans le modèle à l'aide d'un ensemble de séries d'apports (scénarios) observés ou générés aléatoirement représentant des réalisations possibles du processus stochastique sous-jacent. Pour évaluer l'espérance de la fonction de valeur, la probabilité de chaque scénario est calculée par rapport à une variable hydrologique conditionnelle.

D'autres extensions de la PDS, dont la programmation dynamique stochastique duale, sont exposées au chapitre 3.

La PL est parfois étendue au cas stochastique via la *programmation linéaire à deux étapes* où les décisions sont classées en deux groupes : celles qui sont prises avant la réalisation du processus d'apports (première étape) et celles qui sont prises après (deuxième étape ou étape recours). L'incertitude sur les apports peut être modélisée en considérant des distributions de probabilité théoriques ou via des arbres de scénarios (voir chapitre 2).

Par exemple, Seifi et Hipel (2001) utilisent un *programme linéaire stochastique en deux étapes avec recours* pour la planification à long terme d'un système de réservoirs. L'allocation des apports hydrauliques est modélisée avec des pénalités linéaires  $c^+$  associées au surplus  $\pi^+$  et  $c^-$  en cas de rareté  $\pi^-$ . Les décisions de la première étape sont les débits turbinés et les volumes stockés au cours de chaque période. Un ensemble de scénarios sur les apports est considéré dans le programme d'optimisation.

La principale difficulté de la PL stochastique est associée au fait que pour bien représenter l'incertitude sur les apports, il peut s'avérer nécessaire de considérer un grand nombre de scénarios, ce qui résulte en un programme mathématique de grande taille pouvant être difficile à résoudre. Ce défi est parfois contourné via des approches de décomposition telles que la décomposition de Benders et l'agrégation de scénarios par l'algorithme du *progressive hedging*. Une discussion plus approfondie est présentée dans le chapitre suivant.

Dans un modèle avec recours, il est possible qu'une décision prise en contexte d'incertitude ne soit pas réalisable après l'observation des variables aléatoires. C'est pourquoi les violations des contraintes sont parfois pénalisées dans l'objectif (Seifi et Hipel, 2001).

Dans certains contextes, il peut s'avérer plus judicieux d'accepter la possibilité de non réalisabilité, en restreignant la probabilité de cet événement en deçà d'un certain seuil. Par exemple, dans la planification de la production d'énergie électrique, les gestionnaires peuvent spécifier une probabilité de ne pas pouvoir satisfaire la demande (*loss-of-load probability* en anglais) (par ex. un jour sur 10 000) (Sen et Hingle, 1999). Il s'agit de la *programmation avec contraintes probabilistes*

Sreenivasan et Vedula (1996) étudient un modèle d'optimisation d'un réservoir à objectifs multiples afin de maximiser la production d'énergie tout en satisfaisant, à chaque période, la demande en eau pour l'irrigation à un niveau de confiance donné. Les niveaux optimaux du réservoir sont déterminés pour les différents niveaux de confiance associés à la satisfaction de la demande en eau pour l'irrigation.

D'autres exemples plus récents sont présentés dans Ouarda et Labadie (2001), Jiekang *et al.* (2008) et Zeng *et al.* (2014).

Le principal désavantage de la programmation avec contraintes probabilistes est qu'elle produit des solutions « conservatrices » (Strycharczyk et Stedinger, 1987). En effet, il peut être difficile de représenter le risque de violation des contraintes. Celui-ci ne peut être considéré que comme un facteur qui influence l'aversion au risque dans la solution (Labadie, 2004).

### 1.3.3 Méthodes heuristiques

Les méthodes d'optimisation stochastique implicite et explicite (méthodes exactes) sont basées sur des procédures algorithmiques structurées, définies et appliquées sur des données quantitatives. Une méthode heuristique est plutôt basée sur des règles empiriques et appliquée sur des données à la fois qualitatives et quantitatives (Labadie, 2004).

Ces méthodes offrent des alternatives contre les problèmes de dimensionnalité, de non linéarité, d'objectifs multiples, d'incertitude, etc. Cependant, contrairement à la plupart des méthodes exactes, les heuristiques ne garantissent pas la convergence vers des optima locaux. En général, on ne peut s'attendre qu'à des « solutions satisfaisantes ».

Les algorithmes génétiques (AG) sont les méthodes heuristiques les plus fréquentes dans la gestion des réservoirs. Ils font partie de la famille des *algorithmes évolutionnistes* qui se basent sur les notions de la biologie comme la sélection naturelle, la séparation et la mutation pour approximer une solution en considérant une population de potentielles solutions.

Une des principales limites des AG réside dans la difficulté à intégrer de façon explicite des contraintes dans le modèle mathématique, ce qui peut engendrer une grande proportion de solutions irréalisables dans la population. En général, la violation des contraintes est pénalisée dans la fonction objectif.

Dans l'étude de Ahmed et Sarma (2005), la politique d'opération est obtenue à partir de 100 années d'apports naturels synthétiques par un AG. Les solutions sont comparées à celles fournies par un modèle de PDS. Les auteurs concluent que les modèles d'AG sont compétitifs et prometteurs, et peuvent être utilisés pour la détermination des plans d'opération de systèmes de réservoirs. Cependant, ces approches dépendent des applications, donc on n'a aucune base théorique pour conclure à la généralisation de leur applicabilité.

Cai *et al.* (2001) décrivent des stratégies pour résoudre des modèles de réservoirs de grande taille avec des fonctions objectif et de hauteur de chute non linéaires. Les AG sont utilisés pour fixer les « variables complicantes » et linéariser le modèle, puis le programme mathématique est résolu de façon itérative par la PL.

## 1.4 Objet et problématique de la thèse

Cette thèse s’inscrit dans une perspective de gestion à moyen terme des systèmes multi-réservoirs par la programmation stochastique explicite. L’objectif est de déterminer des plans optimaux de turbinage et de stockage dans un contexte d’incertitude sur les apports naturels aux réservoirs. Ainsi, pour bien représenter l’incertitude il faudrait probablement un grand nombre de scénarios sur les apports, ce qui peut faire « exploser » les efforts de calcul. D’où la nécessité de trouver un compromis entre les efforts de calcul et la précision de la politique.

La thèse porte sur l’application de la *méthode d’agrégation de scénarios*, plus particulièrement un algorithme sur lequel repose cette méthode (*Algorithme du Progressive Hedging*) (APH), et de la *programmation dynamique approximative* à ces problèmes de gestion.

Le problème est représenté différemment dans chacun de ces formalismes. Dans l’agrégation des scénarios, les processus stochastiques sont représentés par un arbre d’événements ; une « politique d’opération » est déterminée pour chaque événement. Dans la programmation dynamique, à chaque période, une politique d’opération est calculée pour chaque état possible du système.

L’APH est basé sur un paramètre de pénalité qui joue un rôle important dans sa convergence. En effet, une valeur trop élevée ou trop faible peut entraîner la divergence de l’algorithme. Cependant, il n’existe pas de règle théorique pour choisir une valeur appropriée.

Des valeurs fixes ont été utilisées ou des heuristiques ont été proposées pour mettre à jour la valeur du paramètre selon une trajectoire fixe : croissante ou décroissante. Aucune information sur la convergence globale aux différentes itérations de l’algorithme n’est prise en compte. De plus, ce paramètre est initialisé de façon arbitraire.

Ces constats soulèvent plusieurs questions. Tout d’abord, nous nous demandons s’il ne serait pas plus efficace d’adopter une approche de mise à jour dynamique au lieu



d'utiliser une valeur fixe ou une trajectoire déterminée à l'avance. Deuxièmement, nous nous demandons si de l'information sur la convergence globale de l'algorithme ne peut pas être exploitée pour déterminer le sens de la mise à jour (à la hausse ou à la baisse). Et finalement, ne serait-il pas plus efficace d'initialiser ce paramètre en tenant compte du contexte du problème au lieu de le faire de manière arbitraire ?

La programmation dynamique stochastique est limitée à un petit nombre de réservoirs à cause de sa complexité algorithmique. Malgré cette difficulté, la programmation dynamique est souvent utilisée dans la pratique, de nombreuses compagnies opèrent des systèmes hydroélectriques avec un petit nombre de réservoirs.

La PD classique consiste à discrétiser l'espace des états en une grille régulière (uniforme) et résoudre un problème d'optimisation en chaque point de la grille. Une grille régulière est le produit cartésien de grilles individuelles sur chaque état. Ceci constitue le principal facteur de complexité de la PD. Ensuite, on suppose que les grilles sur chaque axe sont régulièrement espacées, ce qui est un autre facteur de complexité, mais moins important que le premier.

Nous nous demandons s'il est nécessaire d'avoir des grilles très denses pour obtenir une bonne approximation du problème. Ne pourrait-on pas se limiter à résoudre le problème en des points de grille choisis « judicieusement » et faire des interpolations pour des points qui ne sont pas dans la grille ?

Cependant, cette problématique soulève les questions de savoir comment mesurer la qualité de l'approximation et comment choisir ces points de discrétisation. De plus, la mesure de la qualité de l'approximation et la construction de la grille des stocks sont-elles forcément deux étapes distinctes ou peuvent-elles être réalisées conjointement ?

## 1.5 Contributions et organisation de la thèse

Pour répondre à la première partie de la problématique, une approche adaptative de mise à jour du paramètre de pénalité de l'APH est analysée dans le chapitre 2. L'objectif est d'améliorer la convergence de l'algorithme. Nous montrons qu'une approche dynamique améliore le taux de convergence et réduit le nombre d'itérations par rapport à l'approche classique.

Les chapitres 3 et 4 traitent d'une méthode de programmation dynamique approximative qui repose sur le partitionnement de l'espace des stocks en simplexes. Cette

méthode est étudiée en réponse à la deuxième partie de la problématique. La grille construite dans la décomposition simpliciale considère chaque point de grille comme un point dans l'espace global des états plutôt que construit à partir de grilles unidimensionnelles partielles.

Tout d'abord, nous développons un algorithme de décomposition d'hyperrectangles en simplexes dont les sommets forment une grille irrégulière (non uniforme). Ensuite, nous développons une méthode de raffinement de la grille initiale via la division de simplexes existants. Sur chaque simplexe, des bornes sont construites sur la vraie fonction de valeur, ce qui permet de raffiner la grille où cette fonction est le plus incurvée. Une méthode analytique est aussi proposée pour calculer l'espérance de cette fonction dans un contexte information-décision donné.

L'approche de PD est testée dans un contexte industriel. Au chapitre 5, nous présentons une étude de cas de l'entreprise Rio Tinto Alcan. Les caractéristiques du problème sont initialement discutées, puis le problème est modélisé par la programmation dynamique stochastique. Enfin, nous discutons de certains résultats numériques.

Nous concluons la thèse par un résumé et des perspectives de recherche future.

Plusieurs publications scientifiques, dont un article de journal et deux articles publiés dans des actes de conférence, ont été tirées de cette thèse. Un des articles de conférence a reçu le prix du meilleur article étudiant. Le chapitre 2 est essentiellement une traduction d'un article publié dans un journal scientifique. Un manuscrit basé sur l'étude du chapitre 3 a été soumis pour publication dans un autre journal scientifique. Un dernier manuscrit a été tiré des chapitres 4 et 5.

# Chapitre 2

## Méthode d'agrégation de scénarios

### 2.1 Introduction

Comme nous l'avons mentionné dans le premier chapitre, l'objectif de la gestion à moyen terme de la production hydroélectrique est généralement de déterminer des plans de turbinage et de stockage de l'eau pour éventuellement de grands systèmes de réservoirs dans un environnement incertain.

L'obtention de bons plans dépend de deux tâches complémentaires : (i) étant donné une description approximative du processus stochastique, déterminer une politique optimale de turbinage et de stockage (phase d'optimisation) et (ii) déterminer une approximation acceptable du processus stochastique sous-jacent, « acceptable » renvoyant au compromis entre la taille de l'échantillon et l'erreur d'échantillonnage (phase d'estimation). La conjugaison de ces deux aspects reste un grand défi de recherche.

Dans les approches classiques (e.g. Dupačovà *et al.*, 2003), l'estimation et l'optimisation sont soumises à des critères de performance différents. En revanche, l'approche de Kaut et Wallace (2007) propose un critère d'évaluation commun pour les deux phases, ce qui constitue une étape importante vers la cohérence globale.

Pour représenter l'incertitude (sur les apports naturels et éventuellement la demande), une stratégie courante discrétise le processus stochastique sous-jacent via un *arbre d'événements*. Pour décrire ce processus de façon raisonnable, un grand nombre d'*événements* devrait donc être considéré, ce qui peut faire exploser la taille de l'arbre et rendre difficile la résolution du programme d'optimisation.

Cette difficulté peut être contournée en décomposant le programme mathématique glo-

bal en des sous-programmes plus faciles à résoudre. Parmi les méthodes de décomposition, la littérature retient l'« agrégation de scénarios » proposée par Rockafellar et Wets (1991) et peu exploitée dans les modèles mathématiques de gestion des réservoirs. Cette méthode est basée sur un algorithme (*Progressive Hedging*) dont la convergence est réputée lente.

L'objet de ce chapitre est uniquement la phase d'optimisation, supposant qu'une estimation discrète particulière des processus stochastiques des apports naturels et de la demande d'énergie est donnée. Cet intérêt est motivé par les défis actuels confrontés par des compagnies d'hydroélectricité dans plusieurs régions du monde à résoudre des programmes d'optimisation stochastique de grande taille. En particulier, des expériences avec l'algorithme du *progressive hedging* (APH) sont souvent vouées à l'échec à cause de temps de résolution excessifs. Plusieurs idées ont été invoquées afin d'accélérer l'APH, dont l'ajustement du (des) paramètre(s) de pénalité. Cependant, à notre connaissance, aucune n'a conduit à des progrès remarquables.

Ce chapitre examine une version « adaptative » de l'APH comme alternative à la version classique consistant à fixer la valeur du (des) paramètre(s) de pénalité. Nous montrons que l'algorithme classique peut être amélioré en termes de (i) taux de convergence, et de (ii) nombre d'itérations.

La section 2.2 présente une formulation générale du programme d'optimisation des systèmes de réservoirs dans le contexte d'un *arbre d'événements* fini donné. Les sections 2.3 et 2.4 discutent de certaines méthodes de décomposition, dont l'APH. La section 2.5 fait une revue de certaines variantes de cet algorithme ayant pour but d'ajuster le paramètre de pénalité, puis présente notre approche pour initialiser et contrôler ce paramètre. La section 2.6 montre des expériences numériques ainsi que des résultats et analyses. Quelques remarques terminent le chapitre.

## 2.2 Formulation du programme mathématique

Nous considérons un programme d'optimisation hydrothermique multi-réservoirs et multi-étapes. À chaque période, un flux d'eau est tiré des réservoirs et envoyé dans les turbines afin de produire de l'électricité. Nous supposons que chaque réservoir est muni d'un déversoir et le surplus d'eau est délesté dans un réservoir en aval ou hors du système.

L'état du système est décrit par les volumes d'eau dans chaque réservoir à la fin de

la période  $t$ , et les apports naturels à cette même période sont dénotés  $Q_t$ . À chaque période, deux types de décision doivent être prises : les flux d'eau à tirer des réservoirs pour la production d'énergie, notés  $r_t$ , et les flux d'eau à déléster (évacuer), notés  $y_t$ .

Les apports naturels et la demande d'énergie sont incertains. La structure d'information de cette incertitude est décrite par une arborescence donnée, appelée *arbre d'événements* (figure 2.1). Dans cette structure, les variables de décision, les contraintes et la fonction objectif sont divisées en *étapes*.

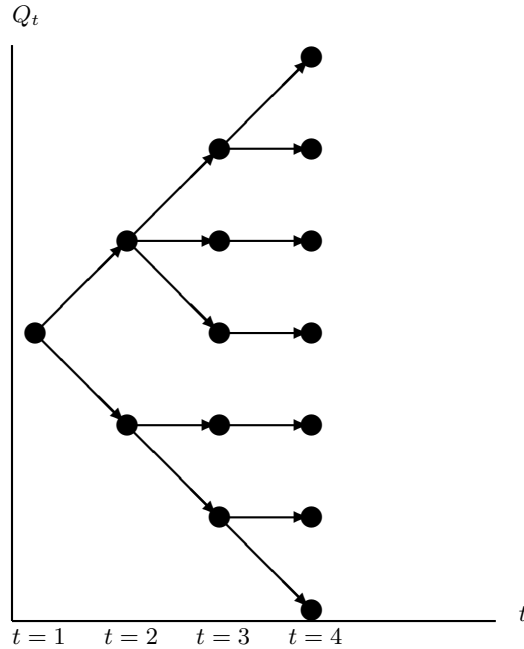


FIGURE 2.1 – Exemple d'un arbre d'événements

Chaque nœud de cet arbre correspond à une réalisation particulière des variables aléatoires, et peut être vu comme un *état d'information* (ou état, pour être plus court). Les états sont indicés par  $k \in K$ . Soit  $a(k)$  l'unique prédécesseur du nœud  $k$  et  $S$  l'ensemble des nœuds terminaux, i.e., les nœuds sans successeurs. Nous notons par  $S_k$  l'ensemble des nœuds terminaux du sous-arbre de racine  $k$ . Nous associons à chaque nœud  $k$ , une période  $t_k$ . Les nœuds terminaux sont associés à la fin de l'horizon  $T$ . Finalement, soit  $K' = \{k \in K \mid |S_k| > 1\}$ , l'ensemble des nœuds qui conduisent à plus d'un nœud terminal.

Le programme d'optimisation global étudié dans ce chapitre cherche à minimiser les coûts d'opération hydrothermiques espérés sur un horizon de planification de  $T$  périodes. Comme dans Pereira et Pinto (1985), à chaque période sont associés un coût

thermique (combustible, etc.) et une pénalité si on n'est pas capable de satisfaire la totalité de la demande (coût de déficit d'énergie). Les paramètres du modèle sont :

$p_k$  : probabilité de l'état  $k$  ;

$d_k$  : demande au nœud  $k$  ;

$Q_k$  : apports naturels au nœud  $k$  ;

$\underline{v}$  : bornes inférieures sur les stocks d'eau ;

$\bar{v}$  : bornes supérieures sur les stocks d'eau ;

$\underline{r}$  : bornes inférieures sur les flux tirés des réservoirs ;

$\bar{r}$  : bornes supérieures sur ces mêmes flux ;

$\vartheta$  : coefficients techniques convertissant les flux en énergie.

Les variables de décision du modèle sont :

$G_k \in R_+$  : production thermique au nœud  $k$  ;

$Z_k \in R_+$  : déficit d'énergie au nœud  $k$  ;

$v_k \in R_+^n$  : stocks d'eau au nœud  $k$  ;

$r_k \in R_+^n$  : flux turbinés au nœud  $k$  ;

$y_k \in R_+^n$  : flux délestés au nœud  $k$ .

Le programme d'optimisation global ( $P$ ) s'écrit alors

$$\begin{aligned} \min \sum_{k \in K} p_k f_k(G_k, Z_k) \\ \text{S.l.c. } G_k + Z_k + \vartheta' r_k = d_k \quad k \in K \end{aligned} \quad (2.1)$$

$$v_k = v_{a(k)} - B r_k - C y_k + Q_k \quad k \in K \quad (2.2)$$

$$\underline{v} \leq v_k \leq \bar{v} \quad k \in K \quad (2.3)$$

$$\underline{r} \leq r_k \leq \bar{r} \quad k \in K \quad (2.4)$$

$$y_k \geq 0 \quad k \in K \quad (2.5)$$

où  $f_k(G_k, Z_k)$  correspond aux coûts d'opération au nœud  $k$  et est supposé être une fonction linéaire des variables  $G_k$  et  $Z_k$ . Les contraintes d'égalité (2.1) et (2.2) sont respectivement l'équation d'équilibre de la demande et de la conservation de l'eau, tandis que les contraintes (2.3) et (2.4) sont des bornes sur les variables de stock et de flux. Les matrices  $B$  et  $C$  sont telles que définies dans le premier chapitre.

## 2.3 Approches de décomposition

La taille du modèle illustré par l'*arbre de scénarios* de la figure 2.1 peut croître de façon exponentielle si beaucoup de réalisations des processus stochastiques sont considérées à chaque étape. Le programme d'optimisation peut donc devenir difficile à résoudre à cause des efforts de calcul nécessaires. Une alternative consiste à décomposer le programme global en des sous-programmes beaucoup plus faciles à résoudre, tout en exploitant la structure du programme original.

Plusieurs approches génériques de décomposition ont été proposées pour les programmes stochastiques multi-étapes. Birge et Louveaux (1997) et Ruszczyński (1999) présentent des revues de ces méthodes ainsi que des perspectives de recherche. Les premiers travaux dans ce domaine s'intéressaient à des variantes de la décomposition de Benders (Benders, 1962), initialement pour des programmes linéaires en deux étapes (e.g. Van Slyke et Wets, 1969), puis des extensions aux cas multi-étapes ont été abordées (e.g. Birge, 1985; Pereira et Pinto, 1991). Après des traitements préalables (*pre-processing* en anglais) appropriés, les méthodes de coupe du type Benders sont considérées très compétitives pour les *programmes linéaires à recours fixe*.

Une deuxième perspective de recherche s'inscrit dans la lignée du travail pionnier de Rockafellar et Wets (1991) sur la *décomposition par scénarios*. Il s'agit d'une approche par dédoublement de variables (*variable splitting* en anglais), où chaque sous-programme serait le programme déterministe qui en résulterait si on pouvait prévoir de façon exacte chaque « scénario », i.e. une réalisation particulière des trajectoires des variables aléatoires.

Les contraintes liantes, dites « non anticipatives », exigent que les décisions des scénarios basés sur un même historique d'information soient identiques. Ces contraintes de non anticipativité sont dualisées et « forcées » par un algorithme de *progressive hedging*. Cette approche est appliquée dans des contextes variés, notamment en finance et logistique (e.g. Mulvey et Vladimirov, 1991; Watson *et al.*, 2008), tandis que les applications dans le contexte de la gestion des réservoirs sont limitées (Dos Santos *et al.*, 2009; Carpentier *et al.*, 2013).

Du même ancêtre, les méthodes de *dédoublement d'opérateur* (*operator splitting* en anglais) appliquent aussi l'*algorithme du point proximal* dans un cadre primal-dual plus général. Dans le contexte de la programmation stochastique multi-étapes, Salinger et Rockafellar (2003) proposent une décomposition par événements (au lieu de scénarios),

ce qui permet d'éviter la duplication de variables, avec un mécanisme de coordination utilisant la récursion efficace de la PD quadratique-linéaire sans contraintes. Des améliorations opérationnelles sont proposées par Pennanen et Kallio (2006). Cependant, un paramètre semblable à un coefficient de pénalité est invoqué. Très peu d'expériences numériques ont été indiquées.

Bien qu'elles ne soient pas strictement des méthodes de décomposition, les méthodes de point intérieur offrent de potentielles alternatives pour les modèles de grande taille. Dans les programmes linéaires stochastiques, le défi est d'exploiter la structure de l'arbre d'événements dans des pas de Newton successifs. Pour les méthodes de point intérieur primales, Blomvall et Lindberg (2002) proposent une approche itérative récursive par une approximation quadratique du pas de Newton.

## 2.4 Décomposition par scénarios et l'algorithme du *progressive hedging*

Dans l'arbre d'événements, un scénario est un chemin orienté de l'état initial vers un état terminal. Ainsi, il y a  $|S|$  scénarios, chacun correspondant à une suite de réalisations possibles des variables aléatoires. Par exemple, dans la figure 2.1, il y a sept scénarios tel qu'illustrés dans la figure 2.2. Les lignes en traitillés connectent les scénarios ayant le même historique jusqu'à la période  $t$ .

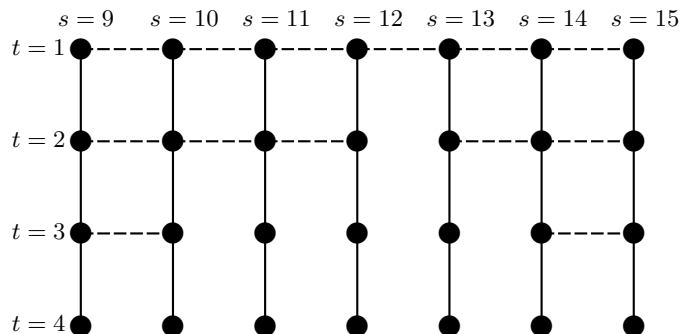


FIGURE 2.2 – Scénarios de l'arbre d'événements de la figure 2.1

La décomposition par scénarios applique la théorie de la dualité pour diviser un programme équivalent ( $P'$ ) en  $|S|$  programmes d'optimisation déterministes, chacun correspondant à un scénario. Elle est une forme de dédoublement de variables où les variables de l'état  $k$  sont dupliquées en  $|S_k|$  copies. Ces variables dupliquées sont maintenant indicées par scénario et période.



Soit la fonction

$$\kappa : S \times T \rightarrow K$$

qui, à tout scénario  $s \in S$  et toute période  $t, 1 \leq t \leq T$ , associe l'état correspondant  $k \in K$ . Il existe un seul état  $k$  tel que  $k = \kappa(s, t)$ . On a donc une correspondance unique entre  $k$  et  $\kappa(s, t)$ .

Dans cette formulation, les variables de décision  $G_k, Z_k, v_k, r_k$  et  $y_k$  associées au programme ( $P$ ) sont remplacées respectivement par

$$\hat{G}_{st} \in R_+ : \text{production thermique à la période } t \text{ sous le scénario } s, 1 \leq t \leq T;$$

$$\hat{Z}_{st} \in R_+ : \text{déficit d'énergie à la période } t \text{ sous le scénario } s, 1 \leq t \leq T;$$

$$\hat{v}_{st} \in R_+^n;$$

$$\hat{r}_{st} \in R_+^n \text{ et}$$

$$\hat{y}_{st} \in R_+^n.$$

De plus, des *variables de consensus*  $w_k \in R_+^n, k \in K'$ , servent à exprimer les contraintes de non anticipativité ( $NA$ ) ci-dessous. La reformulation ( $P'$ ) s'écrit alors

$$\min \sum_{s \in S} p_s \sum_{t=1}^T f_{st}(\hat{G}_{st}, \hat{Z}_{st})$$

$$\text{S.l.c. } \hat{G}_{st} + \hat{Z}_{st} + \vartheta' \hat{r}_{st} = d_{st} \quad s \in S, 1 \leq t \leq T \quad (2.6)$$

$$\hat{v}_{st} = \hat{v}_{s,t-1} - B\hat{r}_{st} - C\hat{y}_{st} + Q_{st} \quad s \in S, 1 \leq t \leq T \quad (2.7)$$

$$\underline{v} \leq \hat{v}_{st} \leq \bar{v} \quad s \in S, 1 \leq t \leq T \quad (2.8)$$

$$\underline{r} \leq \hat{r}_{st} \leq \bar{r} \quad s \in S, 1 \leq t \leq T \quad (2.9)$$

$$\hat{y}_{st} \geq 0 \quad s \in S, 1 \leq t \leq T \quad (2.10)$$

$$\hat{v}_{st} = w_{\kappa(s,t)} \quad \kappa(s,t) \in K' \quad (2.11)$$

Les contraintes (2.6–2.10) sont analogues aux contraintes (2.1–2.5), tandis que les équations (2.11) sont les contraintes de  $NA$  qui assurent que les variables des scénarios appartenant à un même nœud de l'arbre d'événements soient égales. Ces contraintes sont appliquées seulement aux stocks (par les variables  $w_k$ ) étant donné qu'elles s'étendent de façon triviale aux flux. En effet, pour une réalisation particulière des apports, étant donné la relation d'équilibre entre les stocks et les flux (2.7), à l'optimum si les stocks sont égaux, les flux le seront nécessairement.

Considérons le Lagrangien du programme ( $P'$ ) où les contraintes de  $NA$  sont dualisées dans l'objectif :

$$L(X, W, U) = \sum_{s \in S} p_s \left[ \sum_{t=1}^T f_{st}(\hat{G}_{st}, \hat{Z}_{st}) + \sum_{t:\kappa(s,t) \in K'} u_{st}^T (\hat{v}_{st} - w_{\kappa(s,t)}) \right] \quad (2.12)$$

où

$$\begin{aligned}
X &= \{X_s | s \in S\} \\
X_s &= \left\{ \left( \hat{G}_{st}, \hat{Z}_{st}, \hat{v}_{st}, \hat{r}_{st}, \hat{y}_{st} \right) \mid 1 \leq t \leq T \right\}, s \in S \\
W &= \{w_k | k \in K'\} \\
U &= \left\{ u_{st} \mid \kappa(s, t) \in K', \sum_{s: \kappa(s, t) = k} u_{s, t_k} = 0 \forall k \in K' \right\}.
\end{aligned}$$

À (2.12) correspond le programme dual

$$\max_U \text{Inf}_{X, W} L(X, W, U) \quad (2.13)$$

À cause de la grande dimension du vecteur  $U$ , le programme (2.13) peut être difficile à résoudre, et pour des fins de régularisation (convexification), nous définissons le Lagrangien augmenté du programme ( $P'$ ) par

$$\begin{aligned}
\bar{L}(X, W, U, \rho) &= \sum_{s \in S} p_s \left[ \sum_{t=1}^T f_{st}(\hat{G}_{st}, \hat{Z}_{st}) + \right. \\
&\quad \left. \sum_{t: \kappa(s, t) \in K'} \left( u_{st}^T (\hat{v}_{st} - w_{\kappa(s, t)}) + \frac{\rho}{2} \left\| \hat{v}_{st} - w_{\kappa(s, t)} \right\|^2 \right) \right] \quad (2.14)
\end{aligned}$$

où  $\rho$  est un paramètre scalaire de pénalisation.

L'expression (2.14) n'est pas séparable par scénario à cause des variables couplantes  $W$  dans le terme quadratique. Une manière de se défaire de cette difficulté est par une approche de point proximal de minimisation successivement en  $X$ , puis en  $W$ . L'APH de Rockafellar et Wets (1991) est une implantation de cette approche. Telle que décrite dans la table 2.1, chaque itération (numérotée  $\nu$ ) de cet algorithme comprend trois étapes.

Finalement, mentionnons que l'APH peut facilement inclure plusieurs coefficients de pénalité associés à différentes classes de variables. Cette voie de recherche active ne sera pas poursuivie ici.

## 2.5 Ajustement du paramètre de pénalisation

Pour  $\rho$  fixé, dans le cas convexe, l'APH converge vers un optimum global, s'il en existe un. Dans le cas d'un point-col unique, le taux de convergence est linéaire. Dans les autres cas, cet algorithme est un cas d'optimisation par sous-gradient, avec probablement un

TABLE 2.1 – Algorithme du *progressive hedging*

**Étape 1 :** Pour  $W, U$  et  $\rho$  fixés, minimiser le Lagrangien augmenté par rapport à  $X$ , i.e. déterminer

$$X^\nu \in \operatorname{Argmin}_X \left\{ \bar{L}(X, W^{\nu-1}, U^{\nu-1}, \rho) \mid \text{S.l.c.}(2.6) - (2.10) \right\}.$$

Pour  $W$  fixé, le Lagrangien augmenté  $\bar{L}$  est séparable par scénarios, ainsi cette minimisation se traduit par : Pour tout  $s \in S$ , trouver un

$$X_s^\nu \in \operatorname{Argmin}_{X_s} \left\{ \bar{L}_s(X_s, W^{\nu-1}, U_s^{\nu-1}, \rho) \mid \text{S.l.c.}(2.6) - (2.10) \right\}$$

où

$$\begin{aligned} \bar{L}_s(X_s, W^{\nu-1}, U_s^{\nu-1}, \rho) = & p_s \left[ \sum_{t=1}^T f_{st}(\hat{G}_{st}, \hat{Z}_{st}) + \right. \\ & \left. \sum_{t: \kappa(s,t) \in K'} \left( u_{st}^T (\hat{v}_{st} - w_{\kappa(s,t)}) + \frac{\rho}{2} \|\hat{v}_{st} - w_{\kappa(s,t)}\|^2 \right) \right] \end{aligned} \quad (2.15)$$

**Étape 2 :** Pour  $X = X^\nu, U^{\nu-1}$  et  $\rho$  fixés, minimiser le Lagrangien augmenté par rapport à  $W$ . Cela résulte trivialement en

$$w_k^\nu = \frac{\sum_{s \in S_k} p_s \hat{v}_{s,t_k}^\nu}{p_k}, k \in K'$$

**Étape 3 :** Réaliser une itération duale de montée afin de maximiser la fonction duale  $D(U, \rho) \equiv \min_{X,W} \left\{ \bar{L}(X, W, U, \rho) \right\}$  par la mise à jour suivante des variables duales

$$u_{st}^\nu = u_{st}^{\nu-1} + \rho \left( \hat{v}_{st}^\nu - w_{\kappa(s,t)}^\nu \right), \kappa(s, t) \in K'$$

taux de convergence sous-linéaire. Dans plusieurs applications, on mentionne que la convergence de l'algorithme est en effet lente.

Notre étude porte sur la performance de l'algorithme en termes de taux de convergence et de nombre d'itérations. Il est connu que la convergence globale de l'APH est sensible au choix du paramètre de pénalisation. Celui-ci sert de longueur de pas dans la mise à jour des variables duales. Des valeurs élevées de ce facteur peuvent faire osciller les variables duales et causer de « faibles mouvements » dans le terme de pénalité. D'autre part, de faibles valeurs peuvent garantir de bonnes solutions mais peuvent ralentir la

convergence des variables duales (voir [Mulvey et Vladimirov, 1991](#)). Il est clair qu’une valeur appropriée de  $\rho$  dépend du contexte du problème. Il n’existe pas plus de théorie sur ce sujet.

Plusieurs heuristiques ont été proposées pour ajuster ce paramètre. [Watson \*et al.\* \(2008\)](#) associent un coefficient de pénalité différent aux variables des contraintes de NA. Cela leur permet de fixer les valeurs des paramètres en des proportions des coûts unitaires. D’autres auteurs font varier les valeurs du paramètre à chaque itération. Bien que la convergence ne soit plus garantie, ces auteurs ont invoqué des améliorations de la convergence empirique en utilisant des trajectoires de différentes formes de  $\rho$  (dans l’espace des itérations). Ainsi, [Mulvey et Vladimirov \(1991\)](#) relatent une trajectoire concave croissante, tout en indiquant que décroître  $\rho$  proche de l’optimum accélère la convergence.

[Reis \*et al.\* \(2005\)](#) proposent une trajectoire convexe décroissante, où le paramètre est forcé de prendre des valeurs élevées dans les premières itérations pour accélérer la convergence et de faibles valeurs à la fin afin de garantir la « convergence des décisions ». Dans la lignée de [Watson \*et al.\* \(2008\)](#), [Gonçalves \*et al.\* \(2011\)](#) testent à la fois des trajectoires croissante et décroissante dans l’espace des itérations pour mettre à jour les paramètres.

Dans les propositions précédentes, le paramètre de pénalisation est ajusté selon une trajectoire fixe, plutôt que de façon dynamique en fonction de l’état de la solution. Ces trajectoires sont gouvernées par des paramètres qui sont difficilement liés au contexte du problème. Cela renvoie à un « piège » théorique, en ce sens que le Lagrangien augmenté comprend une fonction coût et un terme de pénalité exprimés en des échelles très différentes. Dans l’esprit de l’optimisation bi-critère, on devrait faire attention à toute comparaison inter-échelle.

### 2.5.1 Une heuristique d’apprentissage pour ajuster le paramètre

Nous cherchons à contrôler  $\rho$  dans une perspective d’« apprentissage adaptatif ». À cette fin, nous « surveillons » les indicateurs suivants :

$$\begin{aligned}\theta^\nu &= \sum_{s,t:\kappa(s,t)\in K'} p_s \left\| \hat{v}_{st} - w_{\kappa(s,t)}^\nu \right\|^2 \\ \delta^\nu &= \theta^\nu + \sum_{k\in K'} p_k \left\| w_k^\nu - w_k^{\nu-1} \right\|^2\end{aligned}$$

où  $\nu$  est un compteur d'itérations.  $\theta^\nu$  représente un *écart de non anticipativité* à l'itération  $\nu$ . Si  $\rho$  est constant, la suite  $\{\delta^\nu\}$  converge de façon monotone vers 0 ( $\delta = 0$  est en fait une condition nécessaire et suffisante d'optimalité). Donc  $\delta^\nu$  sera interprété comme un *écart d'optimalité*. Cependant, il convient de mentionner que puisque  $\rho$  varie avec les itérations, la monotonie de la suite  $\{\delta^\nu\}$  n'est plus garantie.

La règle de mise à jour de notre heuristique d'apprentissage est de la forme

$$\rho^\nu = q^\nu \rho^{\nu-1}$$

où la réponse  $q^\nu$  est guidée par deux indicateurs : (i) un indicateur de « progrès » qui reflète un taux de contraction moyen de l'écart d'optimalité, et (ii) un indicateur d'« équilibre » qui reflète la proportion moyenne de l'écart de non anticipativité dans l'écart d'optimalité. Les deux indicateurs sont des nombres purs, ce qui permet d'éviter les pièges de mesure susmentionnés. Une implantation particulière de cette heuristique est résumée dans la table 2.2.

TABLE 2.2 – Heuristique d'apprentissage pour  $\rho$

$\tau^\nu = \frac{\delta^\nu}{\delta^{\nu-1}}$ $\gamma^\nu = \max \{0.1, \min \{0.9, \tau^\nu - 0.6\}\}$ $\sigma^\nu = (1 - \gamma^\nu) \sigma^{\nu-1} + \gamma^\nu \tau^\nu$ $g^\nu = \sqrt{1.1 \sigma^\nu}$
$\alpha^\nu = 0.8 \alpha^{\nu-1} + 0.2 \frac{\theta^\nu}{\delta^\nu}$ $b^\nu = 0.98 b^{\nu-1} + 0.02 \alpha^\nu$ $c^\nu = \max \left\{ 0.95, \frac{1-2b^\nu}{1-b^\nu} \right\}$ $h^\nu = \max \left\{ c^\nu + \frac{1-c^\nu}{b^\nu} \alpha^\nu, 1 + \frac{\alpha^\nu - b^\nu}{1-b^\nu} \right\}$
$q^\nu = (\max \{g^\nu, h^\nu\})^{\frac{1}{1+0.01(\nu-2)}}$ $\rho^\nu = \max \{0.01, \min \{100, \rho^{\nu-1} q^\nu\}\}$

L'indicateur de progrès  $\sigma^\nu$  est un lissage exponentiel non linéaire du progrès instantané  $\tau^\nu = \delta^\nu / \delta^{\nu-1}$ . La formule de lissage est plus réactive quand  $\tau^\nu$  augmente, signalant à temps une mauvaise direction, telle qu'indiquée par la pente croissante dans la figure 2.3. La valeur de cet indicateur est stabilisée (en prenant la racine carrée) afin d'éviter de trop pénaliser la violation des contraintes de NA.

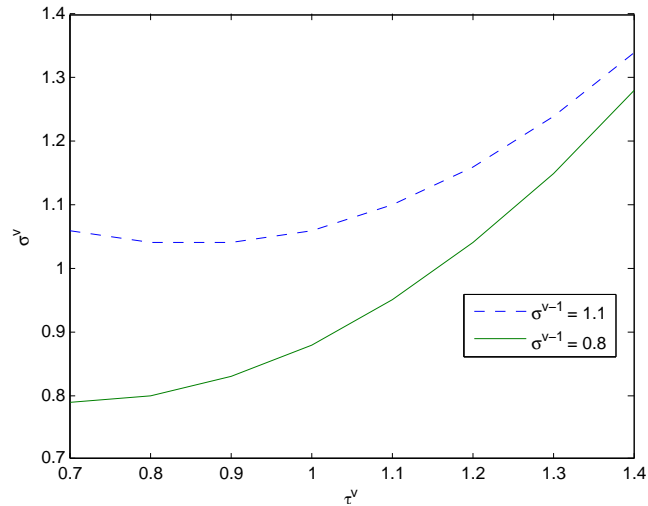


FIGURE 2.3 – L'indicateur de progrès  $\sigma^\nu$  en fonction de  $\tau^\nu$

L'indicateur d'équilibre  $\alpha^\nu$  est comparé à une cible  $b^\nu$  qui représente une proportion « désirée » de l'écart de non anticipativité dans l'écart d'optimalité global. Cette cible, initialisée à 0.5, est ajustée très lentement selon notre expérience de « long terme » des valeurs de  $\alpha$  (initialisé à 1). Une valeur  $\alpha^\nu/b^\nu > 1$  suggère un resserrement des contraintes de NA en augmentant  $\rho$ . Une valeur inférieure à 1 suggère un mouvement opposé, quoique très faible afin d'éviter de l'instabilité dans le programme primal (figure 2.4).

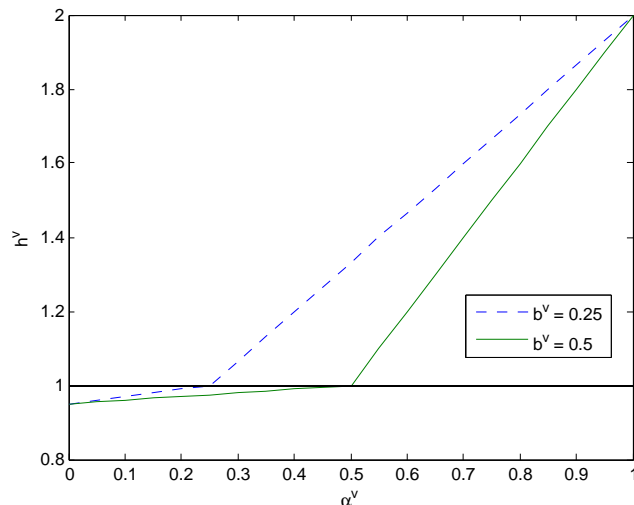


FIGURE 2.4 – L'indicateur d'équilibre stabilisé  $h^\nu$  en fonction de  $\alpha^\nu$

L'influence de l'indicateur d'équilibre sur la réponse  $q^\nu$  se traduit par la fonction linéaire par morceaux  $h(\alpha)$  (figure 2.4). Pour déterminer l'équation de cette fonction, trois situations critiques sont considérées :

- $\alpha = 0$ , i.e. l'écart de non anticipativité est nul, dans ce cas nous fixons la valeur de la fonction à une constante  $c$ .
- $\alpha = b$ , i.e. la cible est atteinte, dans ce cas nous posons  $h(\alpha) = 1$  (Il n'est pas nécessaire de changer la valeur de  $\rho$  si l'écart de non anticipativité n'est pas prédominant.)
- $\alpha = 1$ , i.e. l'écart de non anticipativité représente 100% de l'écart d'optimalité, dans ce cas nous posons  $h(\alpha) = 2$  (la valeur de  $\rho$  est doublée).

L'équation du premier morceau de la fonction  $h(\alpha)$  s'écrit

$$h_1(\alpha) = c + \theta\alpha$$

Quand  $\alpha = b$ , on a  $\theta = \frac{1-c}{1-b}$ . D'où

$$h_1(\alpha) = c + \frac{1-c}{b}\alpha$$

La deuxième partie de la fonction s'écrit

$$h_2(\alpha) = 1 + \theta(\alpha - b)$$

Pour  $\alpha = 1$ , on trouve  $\theta = \frac{1-c}{b}$ . D'où

$$h_2(\alpha) = 1 + \frac{\alpha - b}{1 - b}$$

Remarquons que cette dernière expression s'écrit aussi

$$h_2(\alpha) = \frac{1 - 2b}{1 - b} + \frac{1}{1 - b}\alpha$$

Finalement, mentionnons que la réponse  $q^\nu$  est légèrement influencée par l'itération  $\nu$ , et des bornes, déterminées empiriquement, sont imposées sur la valeur du coefficient de pénalité (voir table 2.2).

## 2.5.2 Initialisation du paramètre de pénalisation

Pour initialiser  $\rho$ , nous considérons une sorte de « décomposition de variance » du Lagrangien augmenté (2.14), qui est la somme des variations dans la fonction coût et

dans la violation des contraintes de NA. Initialisant les variables duales à zéro, cette variation s'écrit

$$\Delta F = \sum_{s \in S} p_s \left[ \Delta \sum_{t=1}^T f_{st}(\hat{G}_{st}, \hat{Z}_{st}) + \frac{\rho}{2} \Delta \sum_{t: \kappa(s,t) \in K'} \left\| \hat{v}_{st} - w_{\kappa(s,t)} \right\|^2 \right]$$

Pour estimer la première partie de l'expression ci-dessus, nous résolvons la version déterministe du programme ( $P$ ) en prenant, à chaque période, la valeur espérée des apports naturels et de la demande d'énergie. Notons par  $E(P)$  la valeur de l'objectif de ce programme déterministe. Pour chaque scénario  $s$ , nous résolvons aussi le programme linéaire

$$\min \left\{ \sum_{t=1}^T f_{st}(\hat{G}_{st}, \hat{Z}_{st}) \mid \text{S.l.c. (2.6) - (2.10)} \right\}$$

La valeur de l'objectif est notée  $O_s$ . Alors,  $\Delta \sum_{t=1}^T f_{st}(\hat{G}_{st}, \hat{Z}_{st})$  est approximé par  $|O_s - E(P)|$ .

Nous approximons  $\Delta \sum_{t: \kappa(s,t) \in K'} \left\| \hat{v}_{st} - w_{\kappa(s,t)} \right\|^2$  par

$$\sum_{t: \kappa(s,t) \in K'} \left\| \hat{v}_{st}^* - w_{\kappa(s,t)}^* \right\|^2$$

où les  $\hat{v}_{st}^*$  sont les solutions des programmes linéaires et les  $w_{\kappa(s,t)}^*$  sont calculés selon l'étape 2 de l'APH. Alors, de façon grossière on a

$$\Delta F \approx \sum_{s \in S} p_s \left[ |O_s - E(P)| + \sum_{t: \kappa(s,t) \in K'} \left\| \hat{v}_{st}^* - w_{\kappa(s,t)}^* \right\|^2 \right]$$

Posons

$$a = \sum_{s \in S} p_s |O_s - E(P)| \text{ et } b = \sum_{s \in S} p_s \sum_{t: \kappa(s,t) \in K'} \left\| \hat{v}_{st}^* - w_{\kappa(s,t)}^* \right\|^2,$$

on a  $\Delta F = a + \frac{\rho}{2} b$ . On pourrait choisir  $\rho$  de telle sorte que

$$\beta a = (1 - \beta) \frac{\rho}{2} b, \quad 0 < \beta < 1,$$

avec  $\beta$  un poids donné.

Par exemple, si  $\beta = 0.3$ , on veut que la variation dans la fonction coût soit 70% de la variation totale. Cette valeur de  $\beta$  sera utilisée dans les expériences numériques. De l'expression ci-dessus, on a

$$\rho = \frac{2\beta a}{(1 - \beta) b}$$

À des fins de stabilité, la dernière formule est modifiée en

$$\rho = \frac{2\beta a}{1 + (1 - \beta) b}$$



## 2.6 Expériences numériques

Nous avons mesuré les performances de l'APH avec paramètres de pénalisation fixe et adaptatif à partir de 60 exemples de problèmes de différentes caractéristiques. Ce cadre expérimental est présenté dans la section 2.6.1. Pour chaque problème, nous avons exécuté la version adaptative de l'algorithme et la version classique (paramètre fixe). Les analyses sont présentées en détail dans la section 2.6.2.

### 2.6.1 Ensemble test

Les 60 problèmes ont été construits par simulation de Monte Carlo sous différentes conditions d'horizon de planification, de configuration de réservoirs, d'arbres d'événements et de réalisation des processus d'apports/demande. Le plan de simulation global est présenté dans la table 2.3 et la figure 2.5.

TABLE 2.3 – Plan de simulation

n	# approx. de scénarios	# de du- plications par cas	# de problèmes
2 et 3	100	5	10
2 et 3	300	5	10
4 et 5	500	5	10
4 et 5	900	5	10
6 et 7	1300	5	10
6 et 7	1800	5	10
Total	—	—	60

Pour simuler les arbres d'événements, nous avons procédé de la manière suivante. Soient  $n$  le nombre de réservoirs,  $n_{st}$  le nombre moyen de successeurs d'un nœud  $k$  à la période  $t$ ,  $K$  l'ensemble des nœuds de l'arbre, et  $K_t$  l'ensemble des nœuds à la période  $t$ . À partir de la deuxième période, pour chaque nœud  $k$  appartenant à l'ensemble  $K_{t-1}$ , nous avons tiré un nombre aléatoire entier de successeurs de ce nœud d'une distribution uniforme sur l'intervalle  $(n_{st} - 1, n_{st} + 1)$ . Les nœuds tirés sont associés à leurs prédécesseurs et rajoutés à la liste  $K_t$ . Cette procédure est résumée dans la table 2.4.

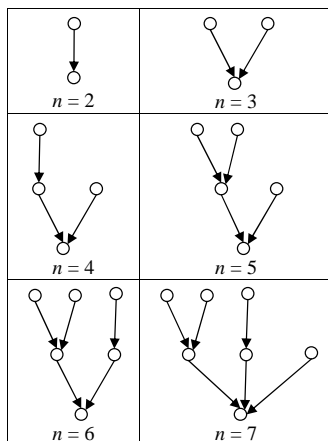
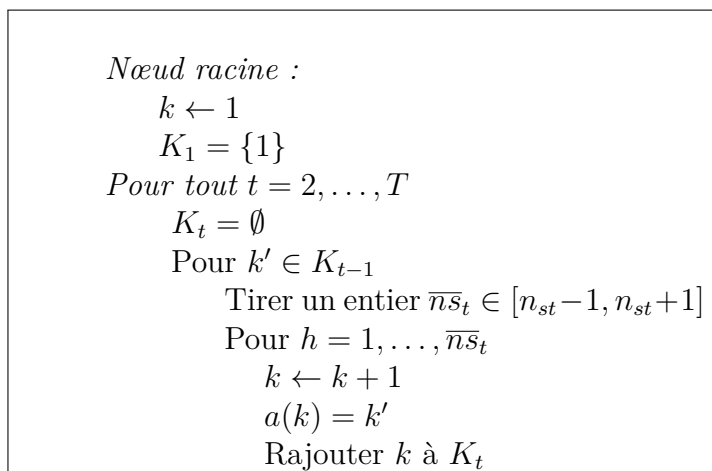


FIGURE 2.5 – Configuration des réservoirs

TABLE 2.4 – Procédure de construction d’un arbre de scénarios



Les réalisations des apports naturels et de la demande ont été tirées de façon indépendante afin de contraster le plus les scénarios. En effet, les contraintes de NA sont d’autant plus difficiles à satisfaire que les scénarios sont contrastés. Si l’algorithme donnait de bons résultats dans ces cas, on pourrait espérer de meilleurs dans des situations plus réalistes avec probablement des corrélations sérielles et spatiales souvent positives. Pour chaque nœud, nous avons tiré des valeurs contrastées représentant des situations avec beaucoup, peu et aucun apport par rapport aux capacités des réservoirs. La table 2.5 résume la démarche.

TABLE 2.5 – Réalisations possibles des apports naturels et de la demande

<b>Apports naturels</b>	
Pour tout $k \in K$	
Pour tout $i = 1, \dots, n$	
Tirer $Q_{ik} =$	$\begin{cases} 1.2\bar{v}_i & \text{avec probabilité } 1/3 \\ 0.6\bar{v}_i & \text{avec probabilité } 1/3 \\ 0 & \text{avec probabilité } 1/3 \end{cases}$
<b>Demande</b>	
Pour tout $k \in K$	
Tirer $d_k =$	$\begin{cases} 1.1 \sum_{i=1}^n \bar{v}_i \vartheta_i & \text{avec probabilité } 1/3 \\ 0.8 \sum_{i=1}^n \bar{v}_i \vartheta_i & \text{avec probabilité } 1/3 \\ 0.1 \sum_{i=1}^n \bar{v}_i \vartheta_i & \text{avec probabilité } 1/3 \end{cases}$

## 2.6.2 Analyses et résultats

Nous avons implanté les deux versions de l'APH, adaptative ( $A$ ) et classique ( $C$ ), avec Matlab 7.11.0, et les programmes quadratiques ont été résolus avec Cplex 12.2. Pour chaque problème  $i$ , nous avons exécuté la version adaptative de l'algorithme en premier, puis utilisé la moyenne des valeurs de  $\rho$  sur l'ensemble des itérations comme valeur fixe de la version classique. Nous estimions qu'il s'agissait d'un bon compromis étant donné qu'il n'y a pas de règle théorique pour choisir ce paramètre.

Pour chaque problème  $i$ , nous avons mesuré le nombre d'itérations  $\nu_i^A$  et  $\nu_i^C$ , et le taux moyen de convergence  $\kappa_i^A$  et  $\kappa_i^C$  défini par  $\kappa_i = (\delta_{i,\nu_i}/\delta_{i,1})^{1/\nu_i}$ . Nous avons utilisé le seuil  $\delta_{i,\nu_i} \leq 0.01$  sur l'écart d'optimalité comme critère d'arrêt.

Définissons

$NS_i$  = nombre de scénarios du problème  $i$ .

$CO_i = \kappa_i^A/\kappa_i^C$  = taux de convergence « relatif » de l'algorithme  $A$  par rapport à l'algorithme  $C$ .

$RN_i = \nu_i^A/\nu_i^C$  = nombre d'itérations « relatif » de l'algorithme  $A$  par rapport à l'algorithme  $C$ .

Afin de pallier de possibles bruits dans les résultats dûs à la simulation de Monte Carlo, nous avons considéré deux étapes : élimination des cas extrêmes et perturbation des

problèmes. Puisque la robustesse est souvent associée à la richesse des scénarios, nous avons construit deux régressions univariées de  $CO_i$  sur  $NS_i$  et de  $RN_i$  sur  $NS_i$ . Les problèmes dont la valeur absolue des résidus était supérieure à  $1.5 \times$  l'erreur standard de l'une des régressions ont été éliminés. Le reste de l'échantillon comprenait 46 problèmes. Les statistiques descriptives consignées dans la table 2.6 montrent que, dans pratiquement tous les cas, la version adaptative de l'algorithme surclasse la version classique à la fois en termes de taux convergence et de nombre d'itérations.

TABLE 2.6 – Statistiques descriptives

<i>Cas</i>	# <i>prob.</i>	# <i>scénarios</i>			$CO_i$			$RN_i$		
		Moyenne	Min	Max	Moyenne	Min	Max	Moyenne	Min	Max
1	4	124.75	90	169	0.9344	0.9186	0.9481	0.3358	0.1705	0.5922
2	5	107.20	84	137	0.9405	0.9103	0.9604	0.3699	0.1502	0.6000
3	5	346.20	308	423	0.9331	0.9035	0.9520	0.3677	0.2317	0.5349
4	5	300.40	246	353	0.9458	0.9214	0.9685	0.3586	0.2355	0.5836
5	5	594.20	495	656	0.9498	0.9270	0.9673	0.4416	0.2637	0.6094
6	4	455.00	348	653	0.8462	0.7816	0.8882	0.1864	0.1631	0.2092
7	5	963.40	807	1139	0.9195	0.9060	0.9455	0.2094	0.1317	0.2954
8	3	851.67	837	876	0.8395	0.7684	0.9546	0.4974	0.3034	0.8000
9	3	1228.00	1148	1295	0.8465	0.8015	0.8750	0.2843	0.1626	0.3939
10	1	1354.00	1354	1354	0.9692	0.9692	0.9692	0.9429	0.9429	0.9429
11	3	1793.33	1540	2022	0.8998	0.8828	0.9136	0.5321	0.3427	0.8077
12	3	1786.67	1611	2024	0.8952	0.8290	0.9543	0.6901	0.3404	1.0000
<i>Total</i>	<i>46</i>	<i>700.20</i>	<i>84</i>	<i>2024</i>	<i>0.9126</i>	<i>0.7684</i>	<i>0.9692</i>	<i>0.3865</i>	<i>0.1317</i>	<i>1.0000</i>

Des tests statistiques indiquent que  $E[CO_i] < 1$  et  $E[RN_i] < 1$  avec des niveaux de signification élevés. Un exemple, parmi tant d'autres, est donné dans la table 2.7.

TABLE 2.7 – Tests de signification de moyennes

Mesure de performance	$CO$	$RN$
Hypothèse nulle $H_0$	$E[CO] \geq 0.93$	$E[RN] \geq 0.46$
statistique t (45 ddl)	-2.3048	-2.3029
Conclusion du test	Rejet de $H_0$ , (valeur $p=1.3\%$ )	Rejet de $H_0$ , (valeur $p=1.3\%$ )

Finalement, nous avons réalisé un autre test de stabilité en perturbant les problèmes. Pour un sous-ensemble aléatoire de 29 des 46 exemples de problèmes originaux, les valeurs des demandes et des apports naturels ont été chacune augmentée ou diminuée de 10% de façon équiprobable. Nous avons réalisé des tests de comparaison de moyennes

avec groupes non homogènes dont les résultats sont montrés dans la table 2.8. L'exposant  $P$  fait référence aux problèmes perturbés.

TABLE 2.8 – Comparaison de performance : problèmes originaux et perturbés

Mesure de performance	$CO$	$RN$
Hypothèse nulle $H_0$	$E[CO] = E[CO]^P$	$E[RN] = E[RN]^P$
statistique t (ddl)	0.6093 (42.23 ddl)	-1.3152 (65.46 ddl)
Conclusion du test	Accepter $H_0$ , ( $p = 54.6\%$ )	Accepter $H_0$ , ( $p = 18.8\%$ )

Les résultats de la table 2.8 confirment la robustesse de nos conclusions initiales. En moyenne, la performance de l'algorithme ne change pas en perturbant les problèmes. Nous pouvons donc raisonnablement conclure qu'une approche adaptative de contrôle du paramètre de pénalisation peut améliorer la performance algorithmique du *progressive hedging* appliqué dans le contexte de la production hydroélectrique.

## 2.7 Conclusion

Nous avons présenté dans ce chapitre de nouvelles heuristiques pour initialiser et mettre à jour le paramètre de pénalisation de l'algorithme du *progressive hedging* dans le contexte de l'optimisation stochastique des systèmes de réservoirs. Contrairement aux approches antérieures, notre heuristique adaptative d'apprentissage met à jour la valeur de ce paramètre suivant une trajectoire dynamique. La règle de mise à jour prend en compte deux considérations : la convergence globale et la réduction de l'écart de non anticipativité.

Des tests réalisés sur 46 problèmes simulés et de taille différente, dans le contexte spécifique de la production d'énergie hydroélectrique, suggèrent qu'une approche de mise à jour dynamique peut améliorer de façon importante le processus de résolution comparativement à une valeur fixe du paramètre. Nous croyons que cette approche peut être facilement étendue aux cas de paramètres de pénalité multiples associés à différentes classes de variables.

Dans les deux chapitres suivants, nous proposons une autre approche d'optimisation stochastique pour aborder la gestion à moyen terme des systèmes de réservoirs.



# Chapitre 3

## Méthode d'approximation simpliciale dans le cadre de la programmation dynamique

### 3.1 Introduction

La programmation dynamique (PD) a été l'une des premières techniques utilisées pour résoudre les problèmes de gestion des réservoirs hydroélectriques. Pour chaque état possible du système, elle cherche le meilleur compromis entre l'utilisation immédiate de l'eau et son utilisation future. À cause de la complexité de cet algorithme, des méthodes approximatives ont été développées afin de réduire les efforts de calcul. Le problème de la dimensionnalité est d'autant plus apparent qu'une grille régulière est utilisée pour chaque variable d'état. La complexité augmente alors de manière exponentielle avec le nombre de variables. À notre connaissance, peu de ces méthodes tiennent compte explicitement de l'erreur d'approximation. Notre objectif est de développer une stratégie d'approximation qui permette de contrôler l'erreur d'approximation.

Nous proposons dans ce chapitre une méthode d'approximation de la fonction de valeur en partitionnant l'espace des stocks d'eau en simplexes. Essentiellement, (i) nous utiliserons une grille irrégulière où l'espacement entre les points dépend de l'incurvation de la fonction de valeur ; (ii) sous des hypothèses de concavité, l'enveloppe concave de cet échantillon de points est une borne inférieure sur la vraie fonction de valeur ; (iii) les facettes de cette fonction de valeur sont des simplexes ; (iv) pour localiser ces points de grille, nous utiliserons ces simplexes au lieu des coordonnées d'une grille régulière, et

(v) utilisant des bornes supérieures sur la fonction de valeur, l'erreur d'approximation indique les régions où la grille doit être raffinée. Ainsi la méthode permet un compromis entre le temps de calcul et le degré de précision souhaité.

Dans la section 3.2, nous présentons une formulation générale du programme mathématique étudié. La section 3.3 est une description des deux principales étapes de la méthode proposée : décomposition d'hyperrectangles en simplexes et division de simplexes. Une approximation concave de la fonction de valeur est exposée dans la section 3.4, tandis que la section 3.5 traite de l'interpolation de cette fonction en des points qui ne sont pas dans la grille. La section 3.6 présente quelques exemples numériques et leurs résultats.

## 3.2 Formulation du programme mathématique

Le modèle général étudié dans ce chapitre est celui présenté à la section 1.2 (chapitre 1), soit :

$$\text{Max } E \left[ \sum_{t=1}^T f_t(v_{t-1}, u_t) + V_{T+1}(v_{T+1}) \right] \quad (3.1)$$

$$\text{S.l.c. } v_t = v_{t-1} - Bu_t - Cy_t + Q_t \quad t = 1, \dots, T \quad (3.2)$$

$$(v_{t-1}, u_t, y_t, v_t) \in D_t \quad t = 1, \dots, T \quad (3.3)$$

Rappelons que  $v_t$  est un vecteur de variables d'état (les niveaux des réservoirs à la fin de la période  $t$ ),  $u_t$  et  $y_t$  sont des vecteurs de variables de décision (respectivement les flux turbinés et les flux délestés), et  $Q_t$  est un vecteur de variables aléatoires (les apports naturels).  $f_t$  représente une fonction de production ou de revenu,  $V_{T+1}(v_{T+1})$  est la valeur terminale de l'eau et  $D_t$  est un ensemble convexe qui contient des bornes sur les variables ou des contraintes couplantes sur les variables d'état et de décision.

Puisque  $f_t$  ne dépend pas de  $Q_t$ , si le processus  $\{Q_t\}$  n'est pas sériellement corrélé, pour  $t = T + 1, T, \dots, 1$ , l'équation de récurrence de la PDS correspondant au programme (3.1–3.3) s'écrit :

$$V_{t-1}(s_{t-1}) = \underset{u_t, y_t}{\text{Max}} \{ f_t(v_{t-1}, u_t) + E_{Q_t} [V_t(v_{t-1} - Bu_t - Cy_t + Q_t)] \} \quad (3.4)$$

$$\text{S.l.c. } (v_{t-1}, u_t, y_t) \in D_t \quad (3.5)$$

où  $V_t$  est la fonction de valeur à la période  $t$ . Étant donné que  $D_t$  est convexe, si  $f_t$  est concave en  $v_{t-1}$  et  $u_t$ , et si  $V_t$  est concave en  $v_t$ , alors  $V_{t-1}$  est concave en  $v_{t-1}$ .



Notre but est de déterminer les fonction  $V_1, \dots, V_T$  pour lesquelles la politique optimale  $(u_t, y_t, v_t)$  peut être calculée au fur et à mesure que les variables aléatoires  $Q_t$  sont observées. Tel que formulé, pour une période  $t$  donnée, ce programme d’optimisation consiste à (i) calculer l’espérance et (ii) déterminer les décisions optimales. Ce chapitre s’intéresse uniquement à l’étape (ii).

À l’exception des rares cas où une forme analytique est disponible, la fonction de valeur doit être évaluée sur un espace d’état discret. L’approche classique discrétise l’espace des états de façon uniforme sur des coordonnées rectangulaires. Plus récemment, des méthodes d’échantillonnage moins régulières ont été considérées dans le but de tenir compte des régions où la fonction de valeur est très variable. Moore (1992) présente plusieurs approches de partitionnement avec changement d’échelle des variables. Munos et Moore (2002) proposent plusieurs raffinements locaux dans une structure représentée par un arbre. Chaque nœud de cet arbre est un hypercube, sauf les nœuds terminaux qui sont partitionnés en simplexes. Krau *et al.* (2014) utilisent ce même partitionnement hiérarchique en hypercubes. Dans ces propositions, les raffinements locaux sont guidés par une mesure générale de l’erreur d’estimation.

L’approche présentée dans ce chapitre est basée sur la partition de l’espace des états, que nous supposons être un hyperrectangle (ci-après pavé plein), en simplexes. La fonction de valeur sera évaluée aux sommets de ces simplexes et interpolée ailleurs. Il s’agit d’une procédure en deux étapes. Tout d’abord, le pavé initial est décomposé en simplexes. Ensuite, les simplexes existants sont itérativement décomposés en sous-simplexes jusqu’à l’atteinte d’un critère de précision fixé au préalable. Des propriétés de concavité sont exploitées pour obtenir des mesures d’erreur d’estimation plus fines (locales). Il s’agit donc d’une méthode adaptative.

### 3.3 Partition de l’espace des stocks en simplexes

Avant d’exposer la méthode, nous commençons par une revue des propriétés des simplexes qui seront invoquées. Rappelons qu’un simplexe dans  $\mathbf{R}^n$  est l’enveloppe convexe d’un ensemble de  $k$  points affinement indépendants<sup>1</sup>. Nécessairement,  $k \leq n + 1$ . Si  $k = n + 1$ , il s’agit d’un simplexe de pleine dimension (dit *plein*), donc d’intérieur non vide.

---

1.  $k$  points sont affinement indépendants si aucun ne se trouve dans l’enveloppe affine engendrée par les  $k - 1$  autres.

### 3.3.1 Représentations d'un simplexe

Nous convenons de la notation suivante : *Conv*, *Int*, *Ri* et *Aff* représentent respectivement l'enveloppe convexe, l'intérieur, l'intérieur relatif et l'enveloppe affine.

Un simplexe dans  $\mathbf{R}^n$  étant un polytope (un polyèdre borné), il peut être représenté comme l'ensemble des combinaisons convexes de ses points extrêmes (ou sommets). C'est une représentation par enveloppe interne. Alternativement, il peut être représenté comme l'ensemble des points satisfaisant un certain ensemble fini d'inégalités. Il s'agit d'une représentation par enveloppe externe.

Considérons  $\{x^1, x^2, \dots, x^{n+1}\}$ , un ensemble de points affinement indépendants dans  $\mathbf{R}^n$ , la matrice  $X = [x^1, x^2, \dots, x^{n+1}] \in \mathbf{R}^{n(n+1)}$  (de rang  $n$ ) et le simplexe plein associé  $S(X) = \text{Conv}\{x^1, x^2, \dots, x^{n+1}\}$ .

**Proposition 3.1.** *Représentation interne :  $S(X)$  est l'ensemble des points  $x \in \mathbf{R}^n$  tels que  $\begin{bmatrix} X \\ e^T \end{bmatrix} \lambda = \begin{pmatrix} x \\ 1 \end{pmatrix}$ ,  $\lambda \geq 0$  a une solution  $\lambda \in \mathbf{R}^{n+1}$ , où  $e = (1, 1, \dots, 1)^T$  est le vecteur-somme et  $e_i$  le  $i^{\text{ème}}$  vecteur unitaire.*

*Démonstration.* Tout point  $x \in \mathbf{R}^n$  tel que  $x \in S(X)$  s'écrit :

$x = X\lambda$ , avec  $\lambda \in \mathbf{R}^{n+1}$  et  $e^T \lambda = 1$ , ce qui revient à :

$$\begin{bmatrix} X \\ e^T \end{bmatrix} \lambda = \begin{pmatrix} x \\ 1 \end{pmatrix}, \lambda \geq 0.$$

□

Les sommets étant affinement indépendants, la matrice  $\begin{bmatrix} X \\ e^T \end{bmatrix}$  est non singulière.

Posons  $[-A \ b] = \begin{bmatrix} X \\ e^T \end{bmatrix}^{-1}$ , où  $A$  est l'opposé de la matrice formée des  $n$  premières colonnes de  $\begin{bmatrix} X \\ e^T \end{bmatrix}^{-1}$ , et  $b$  la dernière colonne de cette même matrice.

**Proposition 3.2.** *Représentation externe :  $S(X)$  est l'ensemble des points  $x \in \mathbf{R}^n$  tels que  $Ax \leq b$ .*

*Démonstration.* De la démonstration précédente on a  $\lambda = \begin{bmatrix} X \\ e^T \end{bmatrix}^{-1} \begin{pmatrix} x \\ 1 \end{pmatrix} \geq 0 \Rightarrow -Ax + b \geq 0$ . □

### 3.3.2 Faces d'un simplexe

$F$  est une face d'un simplexe  $S(X)$  si tout segment de droite de  $S(X)$  qui traverse  $F$  y est entièrement contenu. Toute face d'un simplexe (plein ou non) est aussi un simplexe.

Un simplexe de dimension  $p$  a  $\binom{p+1}{k+1}$  (le nombre de combinaisons de  $k+1$  éléments parmi  $p+1$ ) faces de dimension  $k$ . La table 3.1 en donne des exemples.

TABLE 3.1 – Faces d'un simplexe de dimension  $p$

$k$	Faces
0	$p+1$ points extrêmes
1	$\frac{p(p+1)}{2}$ arêtes
$\vdots$	$\vdots$
$p-1$	$p+1$ facettes
$p$	le simplexe

Soit  $F \subseteq S(X)$  une face de dimension  $k$ . Il existe un sous-ensemble d'indices  $I_F \subseteq \{1, \dots, n+1\}$ ,  $|I_F| = n-k$  tel que  $\forall x \in F$ ,  $\lambda_{I_F} = 0$  et  $A_{I_F}x = b_{I_F}$ . Réciproquement, tout sous-ensemble  $I_F$  ayant ces propriétés définit une face.

Décrivons maintenant la méthode proposée. Tout d'abord, nous discutons de la décomposition d'un pavé en simplexes pleins, puis de la division de simplexes.

### 3.3.3 Division d'un pavé en simplexes

Un pavé plein dans  $\mathbf{R}^n$  est un polyèdre  $H = \{y \in \mathbf{R}^n \mid a \leq y \leq b\}$  où  $a < b$  sont des vecteurs bornés. Sans perte de généralité, par le changement de variable  $x_i = \frac{y_i - a_i}{b_i - a_i}$ , on peut toujours se ramener au pavé unitaire plein  $P = [0, 1]^n$ . C'est ce pavé fixé que nous considérons par la suite.

Le nombre minimal de simplexes dans lequel un pavé peut être découpé est  $n!$ . Le découpage appelé CFK dans ce qui suit est basé sur les travaux indépendants de Coxeter, Freudenthal et Kuhn (Moore, 1992). Cette méthode est utilisée par Munos et Moore (2002) pour découper les hypercubes terminaux en simplexes. Une fois une paire initiale de points opposés choisie, la méthode devient automatique et le découpage est obtenu en considérant tous les chemins simples entre les deux points (voir figure 3.1).

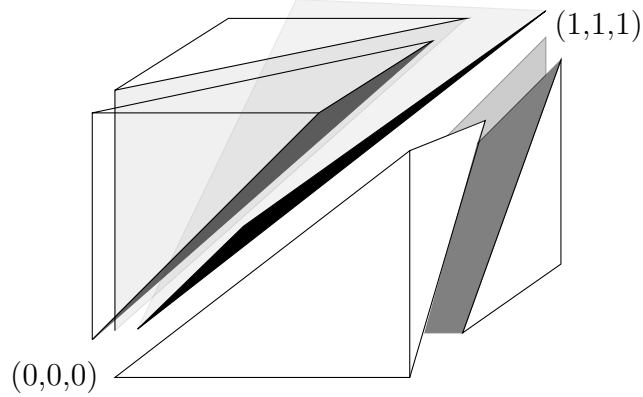


FIGURE 3.1 – Décomposition CFK d'un cube

En termes de précision, cette approche est appropriée pour découper de petits hypercubes locaux. Cependant, dans notre contexte, le découpage de l'hypercube initial est le point de départ pour les opérations ultérieures de division de simplexes. Nous utiliserons des « centres relatifs » comme points de grille initiaux afin d'obtenir une meilleure couverture de l'espace des états. Il n'y a aucune restriction sur la localisation de ces centres relatifs, ce qui offre plusieurs degrés de liberté. La contrepartie de ces avantages sera une plus grande complexité algorithmique.

**Définition 3.1.** *Deux ensembles convexes et bornés seront ici dits séparables si leurs intérieurs relatifs sont disjoints. Une famille d'ensembles convexes bornés est dite séparable si dans cette famille toute paire distincte est séparable.*

**Définition 3.2.** *Soit  $X \subseteq \mathbf{R}^n$  un ensemble convexe. Une collection finie d'ensembles  $C_1, C_2, \dots, C_m$  est un recouvrement de  $X$  si  $\cup_{i=1}^m C_i \supseteq X$ . Cette collection est une quasi-partition de  $X$  si (i) chaque  $C_i$  est un sous-ensemble convexe et borné de  $X$ , et (ii) la collection est séparable.*

Une face de  $P$  de dimension  $k$  est le polyèdre obtenu en fixant  $n - k$  variables à une des deux bornes 0 ou 1. On sait que tout ensemble convexe dans  $\mathbf{R}^n$  peut être partitionné en la collection des intérieurs relatifs de ses faces (Hiriart-Urruty et Lemaréchal, 2001, p.43). La collection des faces de  $P$  est donc séparable.

Soit  $I_0, I_1, I_\nu$ , une tripartition de  $\{1, \dots, n\}$ . Ces ensembles définissent la face de  $P$  de dimension  $|I_\nu|$ :

$$P(I_0, I_1) = \{x \mid x_i = 0 \ \forall i \in I_0, x_i = 1 \ \forall i \in I_1, 0 \leq x_i \leq 1 \ \forall i \in I_\nu\}$$

On n'exclut pas les cas de dimension  $n$  ( $P(\emptyset) = P$ ) ou de dimension 0 (points extrêmes).

**Proposition 3.3.**  $P$  a  $\binom{n}{k}2^{n-k}$  faces de dimension  $k$ . Au total, il y a  $3^n$  faces de toutes dimensions (incluant 0 et  $n$ ).

*Démonstration.* Il y a  $\binom{n}{k}$  façons de choisir  $k$  variables pour former  $I_\nu$  et pour chaque variable restante, deux choix de valeurs possibles, 0 ou 1. Ceci démontre la première affirmation. Le nombre total de faces est donc  $\sum_{k=0}^n \binom{n}{k} 1^k 2^{n-k} = (1+2)^n$ .  $\square$

La table 3.2 en donne des exemples.

TABLE 3.2 – Exemple de nombre de faces de  $P$

n	k								Total
	0	1	2	3	4	5	6	7	
1	2	1							3
2	4	4	1						9
3	8	12	6	1					27
4	16	32	24	8	1				81
5	32	80	80	40	10	1			243
6	64	192	240	160	60	12	1		729
7	128	448	672	560	280	84	14	1	2187

Soit  $P(I_0, I_1)$  une face de  $P$  de dimension  $k$ . On appelle *facette* de  $P(I_0, I_1)$  toute face de  $P(I_0, I_1)$  de dimension  $k-1$ . Elle est obtenue en fixant une variable libre (indiquée dans  $I_\nu$ ) à l'une ou l'autre des bornes 0 ou 1. Par conséquent,  $P(I_0, I_1)$  a  $2k$  facettes. La collection de ces facettes est notée  $\mathcal{F}(I_0, I_1)$ .

Nous décrivons maintenant une procédure itérative de « rehaussement » (*lifting* en anglais) pour quasi-partitionner  $P$  en simplexes. À l'étape  $k$ , considérons toutes les faces de  $P$  de dimension  $k$ . Pour chacune de ces faces  $P(I_0, I_1)$ , supposons qu'il existe une quasi-partition  $\Pi(F)$  pour chaque facette  $F \in \mathcal{F}(I_0, I_1)$  consistant en  $N_{k-1}$  simplexes de dimension  $k-1$ . Choisissons un *centre relatif*  $c(I_0, I_1) \in P(I_0, I_1)$  et construisons la collection :

$$\mathcal{K}(I_0, I_1) = \{Conv(c(I_0, I_1) \cup E) \mid E \in \Pi(F), F \in \mathcal{F}(I_0, I_1)\}$$

**Lemme 3.1.** La collection  $\mathcal{K}(I_0, I_1)$  est une quasi-partition de  $P(I_0, I_1)$ .

*Démonstration.* Tout d'abord, observons que l'ensemble  $Conv(c(I_0, I_1) \cup E)$  est convexe et borné.

Montrons maintenant que  $\mathcal{K}(I_0, I_1)$  est séparable. Soient  $K' = \text{Conv}(c(I_0, I_1) \cup E')$  et  $K'' = \text{Conv}(c(I_0, I_1) \cup E'')$  deux éléments distincts de  $\mathcal{K}(I_0, I_1)$ . Nous montrons que pour tous  $x' \in \text{Ri } K'$ ,  $x'' \in \text{Ri } K''$ ,  $x' \neq x''$ . Étant donné que  $c(I_0, I_1) \in \text{Aff } E'$ , il existe  $\lambda' \in (0, 1)$  et  $y' \in \text{Ri } E'$  tels que  $x' = (1 - \lambda')c(I_0, I_1) + \lambda'y'$ . De même,  $x'' \in (c(I_0, I_1), y'')$  pour  $y'' \in \text{Ri } E''$ . Puisque  $(\text{Ri } E') \cap (\text{Ri } E'') = \emptyset$ , les segments  $(c(I_0, I_1), y')$  et  $(c(I_0, I_1), y'')$  sont disjoints, ce qui prouve que  $x' \neq x''$ . Par conséquent,  $K(I_0, I_1)$  est séparable.

Montrons maintenant que  $\mathcal{K}(I_0, I_1)$  est un recouvrement de  $P(I_0, I_1)$ , i.e. pour tout  $x \in P(I_0, I_1)$ , il existe  $K \in \mathcal{K}(I_0, I_1)$  tel que  $x \in K$ . Ceci est clair si  $x = c(I_0, I_1)$ . Supposons que  $x \neq c(I_0, I_1)$ . Étant donné que  $c(I_0, I_1) \in P(I_0, I_1)$ , la demi-droite  $\{y \mid y = (1 - \lambda)c(I_0, I_1) + \lambda y, \lambda \geq 0\}$  rencontre la frontière relative de  $P(I_0, I_1)$  en un point  $y^\circ$ . Puisque la frontière relative d'un hypercube est la réunion de ses facettes,  $y^\circ$  appartient à au moins une facette  $F^\circ \in P(I_0, I_1)$ . Puisque cette facette est décomposée en une quasi-partition  $\Pi(F^\circ)$ , il existe au moins un élément de cette quasi-partition  $E^\circ \in \Pi(F^\circ)$  tel que  $y^\circ \in E^\circ$ . Ainsi le segment  $(c(I_0, I_1), y^\circ)$  est entièrement contenu dans  $K^\circ \triangleq \text{Conv}(c(I_0, I_1) \cup E^\circ) \in \mathcal{K}(I_0, I_1)$ , et  $x \in K^\circ$ .  $\square$

En résumé, l'itération  $k$  de la procédure de rehaussement débute avec une quasi-partition de chaque face de dimension  $(k - 1)$  en  $N_{k-1}$  simplexes de dimension  $(k - 1)$ . Pour chaque face de  $P$  de dimension  $k$ , nous choisissons un centre relatif pour rehausser chaque simplexe de dimension  $(k - 1)$ , se trouvant sur sa frontière, à un simplexe de dimension  $k$ . Ceci résulte en une quasi-partition de cette face en  $N_k = 2kN_{k-1}$  simplexes. Cette itération se termine quand toutes les faces de dimension  $k$  sont ainsi quasi-partitionnées.

Finalement, comme conditions initiales pour cet algorithme, notons que les points extrêmes de  $P$  sont aussi les faces de dimension 0 ainsi que des simplexes de dimension 0. Ainsi, nous débutons avec des quasi-partitions triviales de ces faces initiales, de sorte que  $N_0 = 1$ . En conclusion :

**Théorème 3.1.** *La procédure de rehaussement découpe le pavé initial  $P$  en  $N_n = 2^n n!$  simplexes.*

Étant donné que  $P$  a  $3^n$  faces (y compris les points extrêmes), et puisque pour chaque face de dimension  $k$ ,  $k = 1, \dots, n$ , nous avons un centre relatif  $c(I_0, I_1)$ , à la fin de l'algorithme de rehaussement, nous avons  $3^n$  points de grille initiaux.

Il est possible de choisir un centre relatif différent pour chaque face de  $P$ . Dans notre implantation, nous choisissons un point  $c \in \text{Int } P$  et définissons  $c(I_0, I_1)$  comme la projection de  $c$  sur  $P(I_0, I_1)$ . La figure 3.2 montre quelques exemples de quasi-partition.

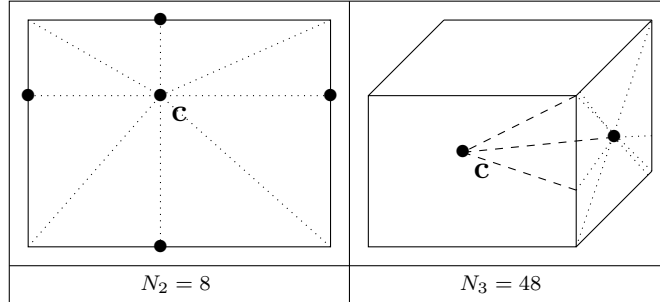


FIGURE 3.2 – Exemples de rehaussement

En termes de complexité relative, l’algorithme CFK découpe l’hypercube  $P$  en  $n!$  simplexes, tandis qu’une grille uniforme, qui divise chaque arête en  $M$  intervalles, décomposerait le pavé  $P$  en  $M^n$  hypercubes. La table 3.3 présente des exemples numériques qui illustrent les ordres de grandeur.

TABLE 3.3 – Comparaison de méthodes de découpage de pavé

$n$	$n!$	$N_n$	$M^n$		
			$M = 2$	$M = 6$	$M = 10$
2	2	8	4	36	100
3	6	48	8	216	1 000
4	24	384	16	1 296	10 000
5	120	3840	32	7 776	100 000
6	720	46 080	64	46 656	1 000 000
7	5 040	645 120	128	279 936	10 000 000
8	40 320	10 321 920	256	1 679 616	100 000 000

Nous avons implanté notre méthode ainsi que l’algorithme CFK avec Matlab *R2012a*, et avons mesuré les temps de calcul, montrés dans la table 3.4, pour  $n = 1$  à 7 réservoirs. Jusqu’à  $n = 4$ , le temps de calcul est pratiquement identique dans les deux algorithmes. Cependant, à partir de  $n = 6$ , la complexité de l’algorithme de rehaussement est largement supérieure à celle de l’algorithme CFK. Cependant, comme nous l’avons indiqué au début, l’avantage de notre méthode est un meilleur échantillonnage de l’espace des états avec au moins  $n$  degrés de libertés.

TABLE 3.4 – Temps de calcul en secondes : rehaussement et CFK

n	CFK	Rehaussement
2	0	0
3	0	0
4	0	0.078
5	0.0312	0.5616
6	0.2028	44.1951
7	1.7316	> 1800

### 3.3.4 Division d'un simplexe

Supposons que l'espace des états ait été quasi-partitionné en simplexes. La fonction de valeur  $V_{t-1}$  ( $V$  dans ce qui suit pour plus de simplicité) a été évaluée en chacun des sommets des simplexes (dans ce qui suit, les points sont repérés en coordonnées naturelles). Notre objectif ici est d'augmenter la précision de cette approximation en augmentant le nombre de points de grille. Ceci sera fait de façon itérative en divisant, un à la fois, les simplexes existants, où chaque division engendre un nouveau point de grille.

Sur les 40 dernières années, la décomposition simpliciale a particulièrement été utilisée dans le domaine de l'optimisation globale (Dickinson, 2013; Paulavičius et Žilinskas, 2013). Cependant, pour autant que nous sachions, les applications dans le domaine de la PDS sont limitées (Munos et Moore, 2002). Moore (1992) présente plusieurs approches de division d'un simplexe. La méthode que nous proposons est basée sur la perspective de raffinement de l'approximation de la fonction de valeur et est guidée par l'incurvation de cette fonction.

Sur un simplexe  $S = S(X)$ , deux types d'information sont disponibles sur la fonction  $V$ . En chaque sommet  $x^i$ , nous avons une évaluation exacte de  $V : z_i = \hat{V}(x^i)$ ,  $i = 1, \dots, n+1$ , et un sous-gradient :  $g^i \in \partial \hat{V}(x^i)$ ,  $i = 1, \dots, n+1$ . Aucune autre information n'est disponible sur cette fonction. Nous construisons deux bornes  $\underline{V}$  et  $\overline{V}$  telles que  $\underline{V}(x) \leq V(x) \leq \overline{V}(x) \forall x \in S$  et  $\underline{V}(x^i) = z_i = \overline{V}(x^i)$ ,  $i = 1, \dots, n+1$ . Une mesure de l'imprécision sur  $V$  en un point  $x \in S$  est donc  $\overline{V}(x) - \underline{V}(x)$ .

Si nous devons évaluer la fonction  $V$  en un nouveau point de  $S(X)$ , il serait préférable que ce soit en un point  $v^0 \in S(X)$  où l'imprécision est maximale, étant donné que notre objectif est d'améliorer l'approximation. Ce point ne peut pas être un sommet de  $S(X)$ , à moins que  $\overline{V}(x) = V(x) = \underline{V}(x) \forall x \in S(X)$ <sup>2</sup>. Ceci entraînera une division de

2. Sur un simplexe, l'interpolation linéaire est unique. Si le point de division est un point extrême,



$S(X)$  en au moins deux sous-simplexes.  $v^0$  sera appelé un *point de division* de  $S(X)$ . L'imprécision maximale  $\Delta = \bar{V}(v^0) - \underline{V}(v^0)$  sera appelée *écart d'approximation* sur  $S(X)$ . La figure 3.3 donne une illustration géométrique de cet écart.

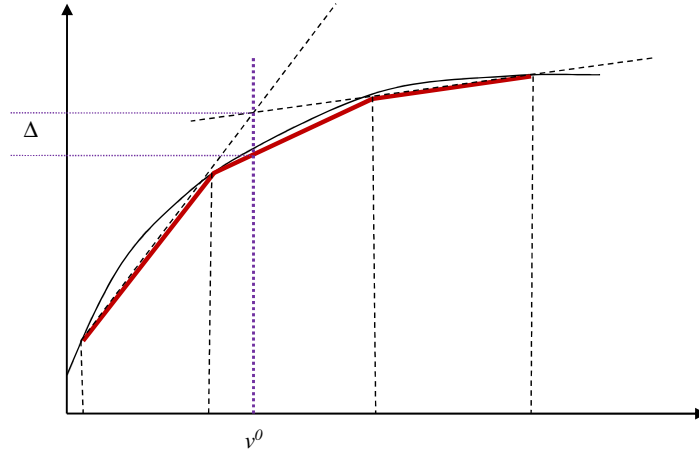


FIGURE 3.3 – Illustration de l'écart d'approximation  $\Delta$

Présentons maintenant les bornes. Puisque la fonction  $V$  est concave, une borne inférieure  $\underline{V}$  est obtenue par interpolation entre les sommets des simplexes :

$$\underline{V}(x) = z^T \begin{pmatrix} X \\ e^T \end{pmatrix}^{-1} \begin{pmatrix} x \\ 1 \end{pmatrix}, \quad x \in S(X), \quad \text{où } z^T = (z_1, \dots, z_{n+1}).$$

Encore par concavité de  $V$ ,  $V(x) \leq z_i + g_i(x^i - x) \quad \forall x, i = 1, \dots, n+1$ , ce qui entraîne la borne supérieure suivante sur  $V$  :  $\bar{V}(x) = \min_{i=1, \dots, n+1} \{z_i + g_i(x^i - x)\}, x \in S(X)$ . Ainsi, l'écart d'approximation  $\Delta$  et le point de division associé  $v^0$  sont donnés par le programme linéaire suivant :

$$\Delta = \underset{\mu, v, \lambda}{Max} \quad \mu - z^T \lambda \quad (3.6)$$

$$S.l.c. \quad \mu \leq z^i + g^i(v - x^i), \quad i = 1, \dots, n+1 \quad (3.7)$$

$$v = X\lambda \quad (3.8)$$

$$e^T \lambda = 1 \quad (3.9)$$

$$\lambda \geq 0 \quad (3.10)$$

La première contrainte calcule le minimum invoqué dans la définition de  $\bar{V}$ . Les autres dans ce cas, la borne inférieure et la borne supérieure coïncident, donc l'erreur d'approximation est nulle.

équations garantissent que le point de division soit une combinaison convexe des sommets du simplexe.

Le point de division  $v^0$  sera un point de l'intérieur relatif  $F^o$  d'une face  $F$  de  $S(X)$  (qu'on notera  $S$  pour simplifier) de dimension  $1 \leq k \leq n$ . Le simplexe  $S$  sera donc divisé en  $k + 1$  sous-simplexes, avec  $v^0$  un point commun. En vue de préserver la symétrie, tout autre simplexe  $S'$  tel que  $S' \cap S = F^o$  sera divisé de façon similaire. En effet, si  $F^o$  avait deux points de division distincts, il serait impossible de recouvrir le partitionnement en simplexes à partir des points de grille (voir figure 3.4 (d)).

La figure 3.4 illustre la division en deux dimensions. Dans le cas (a),  $v^0$  est un point intérieur de  $S$ . Par conséquent, le simplexe est divisé en trois sous-simplexes. Dans le cas (b), le point de division se trouve sur la frontière de l'espace des états, alors le simplexe  $S$  est divisé en deux sous-simplexes. Dans le cas (c),  $v^0$  se trouve sur l'arête commune  $\{x, w\}$  des simplexes  $S$  et  $S'$ ; par conséquent  $S$  et  $S'$  sont chacun divisés en deux sous-simplexes avec  $v^0$  un point extrême commun. Le cas (d) est une situation à éviter.

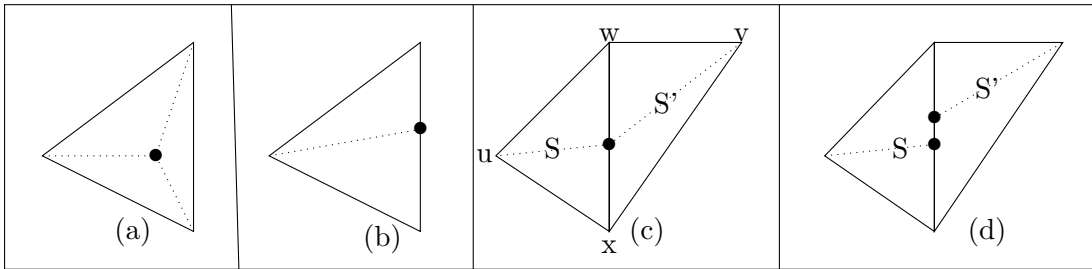


FIGURE 3.4 – Illustration de la division de simplexes

La table 3.5 esquisse un algorithme de division de simplexes. L'algorithme est initialisé avec la liste des sommets du pavé qui servent d'entrées à la décomposition du pavé en simplexes initiaux. Les sommets générés par l'algorithme de décomposition (rehaussement ou CFK par exemple) sont ajoutés dans la liste des sommets initiaux (grille initiale) et la liste des simplexes est initialisée.

Chaque itération de l'algorithme de division de simplexes correspond à un point de grille additionnel ajouté dans la liste des sommets. De même, les nouveaux simplexes sont ajoutés dans la liste des simplexes. Ainsi, les sorties de cet algorithme sont une liste de simplexes et une liste de points de grilles (sommets des simplexes) qui seront utilisés pour approximer la fonction de valeur. La section 3.4 donne plus de détails sur

cette approximation.

Un critère d'arrêt de l'algorithme de division de simplexes peut être soit un nombre fixe d'itérations (ou temps de calcul) ou un seuil sur l'écart d'approximation.

TABLE 3.5 – Algorithme de division de simplexes

<ol style="list-style-type: none"> <li>1. Entrée : un centre relatif du pavé plein unitaire <math>P</math>.</li> <li>2. Initialisation : <ol style="list-style-type: none"> <li>(a) Initialiser la liste des sommets de <math>P</math> : <math>LV = \{x^0, \dots, x^{2^n-1}\}</math>.</li> <li>(b) Décomposer le pavé initial en simplexes (CFK ou rehaussement), mettre à jour la liste des sommets <math>LV</math> et initialiser la liste des simplexes <math>LS</math>.</li> <li>(c) Pour chaque point de la grille initiale <math>x^i</math>, calculer la fonction de valeur <math>z_i = V(x^i)</math> et un sous-gradient <math>g^i(x_i)</math>.</li> <li>(d) Pour chaque simplexe <math>S_i</math> dans <math>LS</math>, calculer son <math>\Delta_i</math> et son <math>v_i^0</math> en résolvant le problème (3.6–3.10).</li> <li>(e) Mettre <math>k = 1</math>.</li> </ol> </li> <li>3. Itération <math>k</math> : <ol style="list-style-type: none"> <li>a) Choisir un simplexe non divisé <math>S</math> dont l'écart d'approximation est maximal et de point de division <math>v^0</math>; calculer <math>V(v^0)</math> et <math>g(v^0)</math>.</li> <li>b) Diviser <math>S</math> au point <math>v^0</math> et le marquer comme divisé. Ajouter chaque sous-simplexe <math>E_i</math> à <math>LS</math>, puis calculer son <math>\Delta_i</math> et son <math>v_i^0</math> en résolvant le programme (3.6–3.10). Ajouter <math>v^0</math> à la liste <math>LV</math>.</li> </ol> </li> <li>4. À moins qu'un critère d'arrêt soit atteint, incrémenter <math>k</math> et retourner à l'étape 3.</li> </ol>
---

La figure 3.5 montre un exemple de division obtenu après 76 itérations avec une précision  $\Delta = 10$  pour la fonction fictive :  $V(v) = 1200 - \frac{1}{200}(\bar{v} - v)^T \begin{pmatrix} 3 & 2 \\ 2 & 4 \end{pmatrix} (\bar{v} - v)$ , dans l'espace  $(0, 0)^T \times (200, 100)^T$ , où  $\bar{v} = (200, 100)^T$ .

### 3.4 Approximation de la fonction de valeur

La méthode de raffinement de la grille proposée dans la section précédente repose sur l'hypothèse de concavité de la fonction  $V_{t-1}$ . Dans cette section, nous présentons une approximation concave de cette fonction. Par simplicité, nous considérons un problème de réservoirs simple sans effets de hauteur de chute. Cependant, nous croyons que l'approche peut être étendue pour prendre en compte de tels effets.

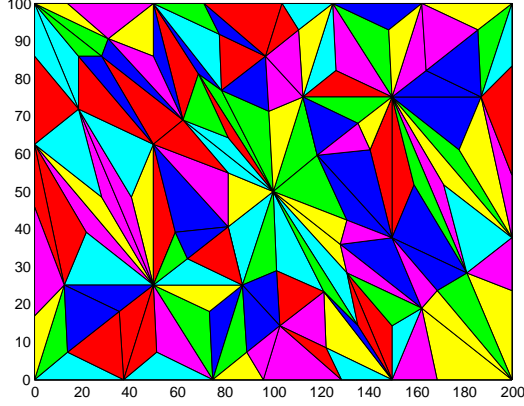


FIGURE 3.5 – Exemple de partitionnement de l'espace des stocks en simplexes

Pour  $t = T + 1, \dots, 1$ , soit la récurrence de PDS suivante associée au programme (3.4-3.5) :

$$V_{t-1}(v_{t-1}) = \underset{u_t, y_t}{Max} \{f_t(v_{t-1}, u_t) + E_{Q_t} [V_t(v_t)]\} \quad (3.11)$$

$$\text{S.l.c. } v_t = v_{t-1} - Bu_t + Q_t - Cy_t \quad (3.12)$$

$$\underline{v} \leq v_t \leq \bar{v} \quad (3.13)$$

$$\underline{u} \leq u_t \leq \bar{u} \quad (3.14)$$

$$y_t \geq 0 \quad (3.15)$$

où  $\underline{v}$  et  $\bar{v}$  sont respectivement des bornes inférieures et supérieures sur les niveaux des réservoirs.  $\underline{u}$  et  $\bar{u}$  sont des bornes inférieures et supérieures sur les flux turbinés; les matrices  $B$  et  $C$  sont telles que définies dans le premier chapitre.

Soit  $\rho_t = v_{t-1} - Bu_t - Cy_t$ , les stocks d'eau avant l'observation des apports naturels. Dans ce qui suit, par simplicité, nous omettons toute référence à l'indice de temps. Soit  $V_{t-1}(v) = V(v)$ ,  $E_{Q_t} [V_t(v_t)] = E_{Q_t} [V_t(\rho_t + Q_t)] = W_t(\rho_t) = W(\rho)$ . Sous l'hypothèse d'absence de hauteur de chute, notons par  $f_{it}(v_{i,t-1}, u_{i,t}) = f_i(u_i)$ ,  $i = 1, \dots, n$ , la fonction de production au site  $i$ . Soit  $f(u) = \sum_{i=1}^n f_i(u_i)$ .

Les quantités  $u$  et  $\rho$  sont continues, il est donc impossible de calculer  $W$  et  $f$  pour chaque  $\rho$  et  $u$  respectivement. À la période précédente ( $t$ ), supposons que nous avons construit une grille de stocks  $\{\hat{\rho}_j \mid j \in J_v\} \subseteq [\underline{v}, \bar{v}]$  en utilisant l'algorithme de division de simplexes (table 3.5 plus haut), et supposons que la fonction  $W$  soit connue en chaque point de cette grille.

Supposons aussi que la fonction  $f$  soit connue sur un échantillon donné de débits  $\{\hat{u}_i \mid i \in I_v\} \subseteq [\underline{u}, \bar{u}]$ . Un exemple concret est présenté dans le chapitre 5.  $I_v$  et  $J_v$  sont des ensembles d'indices servant à distinguer les points de la grille des stocks en fonction des réservoirs et les points de la grille des débits en fonction des centrales.

Une approximation concave et affine de la fonction de valeur  $V$  est la valeur optimale du programme linéaire paramétrique suivant :

$$\hat{V}(v) = \underset{\rho, u, y, \mu, \lambda}{Max} \left\{ \sum_{i \in I_v} f(\hat{u}_i) \mu_i + \sum_{j \in J_v} W(\hat{\rho}_j) \lambda_j \right\} \quad (3.16)$$

$$\text{S.l.c. } \rho = v - Bu - Cy \quad (3.17)$$

$$u = \sum_{i \in I_v} \hat{u}_i \mu_i \quad (3.18)$$

$$\rho = \sum_{j \in J_v} \hat{\rho}_j \lambda_j \quad (3.19)$$

$$\sum_{i \in I_v} \mu_i = 1 \quad (3.20)$$

$$\sum_{j \in J_v} \lambda_j = 1 \quad (3.21)$$

$$\mu, \lambda \geq 0 \quad (3.22)$$

Les contraintes 3.18 indiquent que les débits turbinés sont interpolés à partir des débits de la grille  $\{\hat{u}_i \mid i \in I_v\}$  (ils sont une combinaison convexe des points de cette grille). De même, par les contraintes 3.19, les stocks après turbinages ( $\rho$ ) sont interpolés à partir des stocks de la grille  $\{\hat{\rho}_j \mid j \in J_v\}$ . Les contraintes (3.20–3.22) indiquent que  $\mu$  et  $\lambda$  sont des vecteurs de coefficients de combinaisons convexes. Ainsi, dans l'objectif, la production est interpolée à partir des évaluations des fonctions de production sur la grille  $\{\hat{u}_i \mid i \in I_v\}$  et la fonction  $W$  à partir de ses évaluations sur la grille  $\{\hat{\rho}_j \mid j \in J_v\}$  (équation 3.16).

Étant donné que les points  $\{\hat{\rho}_j \mid j \in J_v\}$  sont choisis dans les limites  $[v, \bar{v}]$ , les contraintes (3.13) deviennent redondantes. Il en est de même des contraintes (3.14)<sup>3</sup>.

Dans ce contexte :

**Proposition 3.4.** *La fonction  $\hat{V}$  est concave en  $v$ .*

*Démonstration.* Le domaine réalisable (3.17 – 3.22) est convexe. L'objectif est linéaire en  $(\mu, \lambda)$  et est maximisé. Il s'ensuit que la fonction  $\hat{V}$  est concave en  $v$ .  $\square$

<sup>3</sup>. Toutefois, les points de grille pourraient être non réalisables. Auquel cas les contraintes (3.13) et (3.14) devraient être réintroduites.

Finalement, mentionnons qu'à l'étape 2c et 3a de l'algorithme de division de simplexes (voir table 3.5), pour tout point  $v$ , les multiplicateurs de Lagrange associées aux contraintes (3.17) donnent un sous-gradient  $g(v)$ .

### 3.5 Interpolation de la fonction de valeur

Cette section sera utile pour la méthode de calcul de l'espérance de la fonction de valeur exposée dans le chapitre suivant (section 4.4). Elle ne sera pas utilisée dans le reste de ce chapitre. Nous la gardons ici étant donné qu'elle fait suite aux deux sections précédentes.

À chaque période  $t$ , nous disposons de l'évaluation de la fonction  $\hat{V}_t$  en chaque point d'une grille  $\{\hat{\rho}_{jt} | j \in J_t\}$  (sommets des simplexes). Dans certaines situations, nous pourrions avoir besoin de l'évaluation de cette fonction en des points qui ne sont pas dans cette grille.  $\hat{V}_t$  sera interpolée en ces points à partir des sommets des simplexes. Par conséquent, nous devons tout d'abord identifier les simplexes non divisés qui contiennent ces points.

Étant donné que certains simplexes sont divisés en sous-simplexes, la partition globale de l'espace des états a une structure imbriquée, ce qui peut être représentée par une arborescence. Les nœuds de cette arborescence sont des simplexes. Ses arcs représentent des divisions. Un simplexe qui n'est pas divisé (une feuille dans l'arborescence) sera dit *actif*. L'ensemble des simplexes actifs forme la partition la plus fine de l'espace des stocks. Ce sera notre ensemble d'intérêt. Cependant, nous garderons en mémoire tous les simplexes afin de localiser plus efficacement un point quelconque de l'espace des états.

Nous voyons facilement qu'un point  $x \in S(X)$  si et seulement si  $\lambda = \begin{pmatrix} X \\ e^T \end{pmatrix}^{-1} \begin{pmatrix} x \\ 1 \end{pmatrix} \geq 0$ . Pour localiser le plus petit simplexe contenant  $x$ , on peut appliquer une procédure de recherche en profondeur dans l'arborescence. La logique de recherche, résumée dans la table 3.6, se base sur le fait que si le point  $x$  appartient à un simplexe divisé (*enfant*), il appartient également à son *père*.

Il se peut qu'on ait besoin également de localiser une suite de points  $\langle x_{kt} | k = 1, \dots, w \rangle$  le long d'une certaine trajectoire donnée et d'interpoler  $\hat{V}_t$  en chaque point de la suite. Ceci sera fait à l'aide de notre dernière opération appelée *traversée de simplexes*. Partant d'un point  $x \in S(X)$ , notre but est d'aller le plus loin possible dans une direction donnée  $\delta \in \mathbf{R}_+^n \setminus \{0\}$ , sans sortir du simplexe.

TABLE 3.6 – Algorithme de localisation d'un point

<p>Initialisation : Trouver quel simplexe de la liste des simplexes initiale contient <math>x</math>.</p> <p>Ensuite, boucle de recherche (descente dans l'arbre) :</p> <p style="padding-left: 40px;">Tant que père divisé</p> <p style="padding-left: 80px;">pour <math>j \in \text{Enfants}(\text{Père})</math></p> <p style="padding-left: 120px;">Si test d'appartenance positif, père = <math>j</math>.</p>
---

Nous disons que le simplexe  $S(X)$  *contient proprement* le point  $x$  dans la direction  $\delta$ , ce que nous notons par  $x \in_\delta S(X)$ , si un pas de déplacement positif est possible, i.e. si pour tout  $\varepsilon > 0$  suffisamment petit,  $x + \varepsilon\delta \in S(X)$ . Il s'ensuit de cette définition que  $x \in_\delta S(X)$  si et seulement si  $\forall j = 1, \dots, n+1, \lambda_j = 0 \Rightarrow \sigma_j \geq 0$ , où  $x \in S(X), \lambda = \begin{pmatrix} X \\ e^T \end{pmatrix}^{-1} \begin{pmatrix} x \\ 1 \end{pmatrix}$  et  $\sigma = \begin{pmatrix} X \\ e^T \end{pmatrix}^{-1} \begin{pmatrix} \delta \\ 0 \end{pmatrix}$ . Cette propriété tient, car par définition,  $x \in_\delta S(X) \Leftrightarrow \exists \varepsilon > 0$ , tel que  $\lambda + \varepsilon\sigma \geq 0$ .

Le déplacement maximal nous amènera à un point  $x' \neq x$  situé à la frontière de  $S(X)$  et tel que  $x' \notin_\delta S(X)$ . Formellement, nous cherchons le pas de déplacement

$$\theta^* = \max\{\theta \in \mathbf{R}_+ \mid x + \theta\delta \in S(X)\}$$

**Proposition 3.5.**  $\theta^* = \min \left\{ -\frac{\lambda_j}{\sigma_j} \mid \sigma_j < 0 \right\}, j = 1, \dots, n+1$ .

*Démonstration.*  $\theta^*$  doit satisfaire  $\hat{\lambda}(\theta^*) = \lambda + \theta^*\sigma \geq 0$ . Alors  $\theta^* = \max\{\theta \mid \theta\sigma \geq -\lambda\}$ . Si  $\sigma_j \geq 0, j = 1, \dots, n+1$ ,  $\theta\sigma \geq -\lambda_j \forall \theta \geq 0$ . Pour  $\sigma_j \leq 0$ , on voit que  $\theta \leq -\frac{\lambda_j}{\sigma_j}$ . On a donc  $\theta^* = \min \left\{ -\frac{\lambda_j}{\sigma_j} \mid \sigma_j < 0 \right\}$ . □

La traversée de simplexes est illustrée dans la figure 3.6.

L'opération de traversée peut être implantée par l'algorithme décrit dans la table 3.7 ci-dessous. À chaque itération  $k$ , en plus de calculer le pas de déplacement  $\theta^*$ , la fonction de valeur est interpolée ( $\omega_k = \hat{V}_t(x_k)$ ) aux différents points de la suite à partir de ses évaluations aux sommets des simplexes (points de la grille). Les conditions initiales de cet algorithme sont un point initial  $x_0$ , une direction  $\delta$  et une zone cible à atteindre  $Z$ . Ces différents éléments seront contextualisés dans le prochain chapitre (section 4.4).

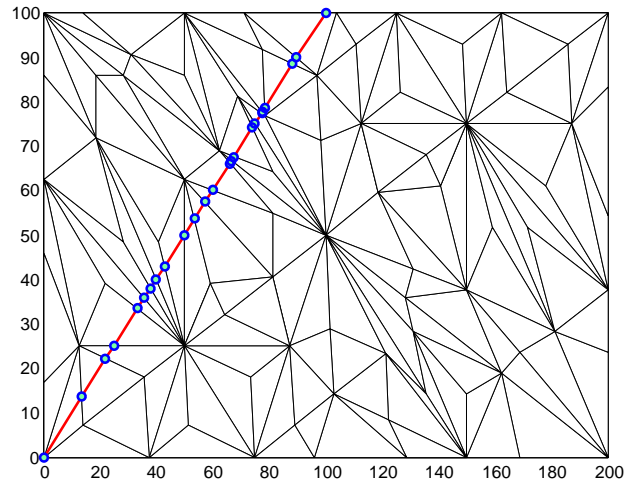


FIGURE 3.6 – Exemple de traversée de simplexes

TABLE 3.7 – Algorithme de traversée de simplexes dans une direction donnée

Initialisation :

Données :  $x_0, \delta$  et  $Z$

Trouver un simplexe  $S_0$  tel que  $x_0 \in \delta S_0$

$k \leftarrow 1$

$LN = \{x_0\}$  et  $LV = \{\omega_0\}$ .

Itération  $k$

Tant que  $x_k \notin Z$

$\theta^* = \max\{\theta \mid (x_k + \theta\delta_k) \in S_k\}$

$x_{k+1} = x_k + \theta^*\delta$

$\omega_{k+1} = \hat{V}_l(x_{k+1})$  (Par interpolation à partir des sommets de  $S_k$ )

Ajouter  $x_{k+1}$  à la liste  $LN$  et  $\omega_{k+1}$  à  $LV$ .

Si  $x_{k+1} \in Z$ , sortir

Sinon

Trouver un simplexe  $S_{k+1}$  tel que  $x_{k+1} \in \delta S_{k+1}$

$k \leftarrow k + 1$

Fin\_Si

Fin\_Tant que



## 3.6 Expériences numériques

Nous avons testé la performance de la méthode simpliciale à l'aide de plusieurs exemples de problèmes simulés, et sous diverses configurations de réservoirs. La méthode a été implantée avec Matlab R2012 a. Les programmes linéaires ont été résolus avec Cplex 12.5.1.0. Pour chaque configuration, nous avons exécuté l'algorithme de rehaussement et enregistré la liste des simplexes initiaux.

Le centre relatif global  $c$  a été choisi comme le point où l'écart d'approximation est maximal sur le pavé initial. Le programme (3.6–3.10) a donc été résolu en utilisant les  $2^n$  sommets du pavé comme points de tangence. Cependant, dans les cas où le centre relatif était sur une frontière, le barycentre des sommets du pavé était choisi comme centre relatif. Pour chaque problème test, nous avons exécuté une récurrence du programme (3.16–3.22). Le cadre expérimental est présenté dans la section 3.6.1, et les résultats sont discutés dans la section 3.6.2.

### 3.6.1 Cadre expérimental

Nous avons choisi les quatre configurations de réservoirs montrés dans la figure 3.7.

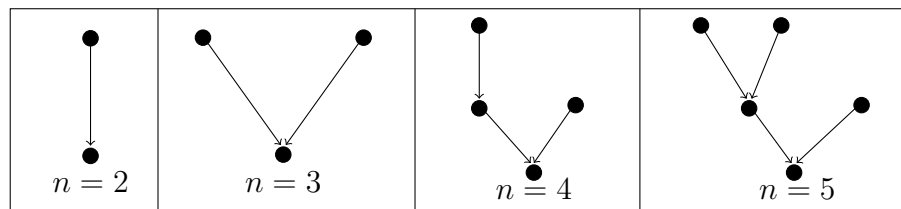


FIGURE 3.7 – Configurations de réservoirs

Pour chaque configuration, nous avons réalisé des simulations de Monte Carlo sur les paramètres suivants : les bornes sur les niveaux des réservoirs  $\underline{v}_i$  and  $\bar{v}_i$ , les bornes sur les flux d'eau turbinés  $\underline{u}_i$  et  $\bar{u}_i$  et les apports naturels  $q_i$ ,  $i = 1, \dots, n$ . Les valeurs des paramètres ont été tirés de façon aléatoire et indépendante de distributions uniformes de supports de la forme  $[a, b]$  spécifiques à chaque type de paramètres mais identiques pour tous les réservoirs. La table 3.8 résume les valeurs numériques de ces bornes.

À des fins de simulation, nous n'avons pas calculé l'espérance sous-jacente de la fonction  $W$ , cependant afin de prendre en compte les effets des apports naturels, nous avons utilisé leur espérance  $q = E[Q]$ . Pour chaque site, nous avons donc supposé que l'apport simulé correspondait à l'espérance du processus stochastique sous-jacent.

TABLE 3.8 – Bornes numériques sur les paramètres des modèles

Paramètre	$a$	$b$
$\underline{v}_i$	500	2000
$\bar{v}_i$	5000	15000
$\underline{u}_i$	0	0
$\bar{u}_i$	5000	20000
$q_i$	0	3000

La fonction  $W$  était de la forme

$$W(v) = m - \frac{1}{2}(v + q - \theta)^T A(v + q - \theta), \quad m > 0$$

où  $A$  est une matrice semi-définie positive de dimension  $n \times n$ , et  $\theta$  un vecteur choisi dans le voisinage de  $\bar{v}$ . La fonction  $W$  est concave en  $v$ .

Le nombre de duplications réalisées pour chaque configuration est consigné dans la table 3.9.

TABLE 3.9 – Nombre d'exemples de problèmes

# de réservoirs	# de duplications
2	15
3	15
4	10
5	5
# total de problèmes	45

Pour chaque site  $i$ , la fonction de production était de la forme

$$f_i(u_i) = \beta_i[(u_i + \gamma_i)^{\alpha_i} - \gamma_i^{\alpha_i}], \text{ avec } 0 \leq \alpha_i \leq 1, \beta_i \geq 0, \gamma_i \geq 0$$

Si  $\beta_i > 0$ , la fonction  $f$  est concave et croissante. Les fonctions  $f$  et  $W$  ont été utilisées pour approximer la fonction de valeur comme nous l'avons décrit dans la section 3.4.

Sur les duplications, les paramètres  $\beta_i, \gamma_i, \alpha_i$  et  $m$  étaient fixes ;  $A$  et  $\theta$  étaient générés aléatoirement.

Pour chaque configuration de réservoirs, la table 3.10 présente les caractéristiques des problèmes résolus pour chaque récurrence de la PD.

De plus, à chaque itération de l'algorithme de division de simplexes, pour calculer le point de division  $v^0$  et l'écart d'approximation  $\Delta$ , un programme linéaire ayant  $2n + 2$

TABLE 3.10 – Caractéristiques des problèmes

$n$	# de variables	# de contraintes
2	78	90
3	81	98
4	84	106
5	87	114

variables et  $3n + 3$  contraintes a été résolu. Pour chaque duplication, au total, le temps de calcul variait approximativement entre 2 et 30 minutes.

### 3.6.2 Résultats

Pour chaque configuration de réservoirs, nous avons calculé la moyenne de l'écart d'approximation ( $\Delta$ ) (sur les duplications) à chaque itération ainsi qu'un intervalle de confiance à 95% sur la vraie moyenne. La figure 3.8 montre un exemple de grille en dimension 2. Il convient de noter l'imprécision qui résulterait si une grille régulière était utilisée. Peu de points de discrétisation ont été choisis dans la région supérieure des stocks, signalant ainsi une région de plus faible incurvation de la fonction de valeur.

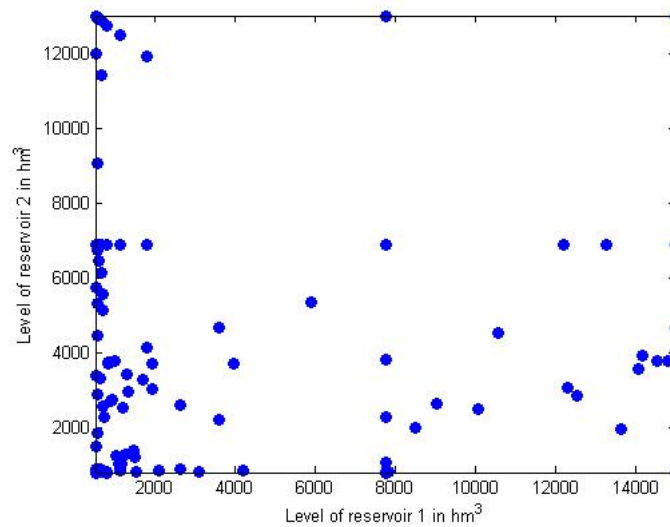


FIGURE 3.8 – Exemple de grille pour un modèle à deux réservoirs

La figure 3.9 montre l'évolution de la valeur moyenne de l'erreur relative et l'intervalle de confiance associé sur les itérations pour chaque configuration de réservoirs. Au début,

l'erreur relative se contracte de façon importante alors qu'elle décroît lentement par la suite. Nous conjecturons que le taux de convergence de l'algorithme est sous-linéaire.

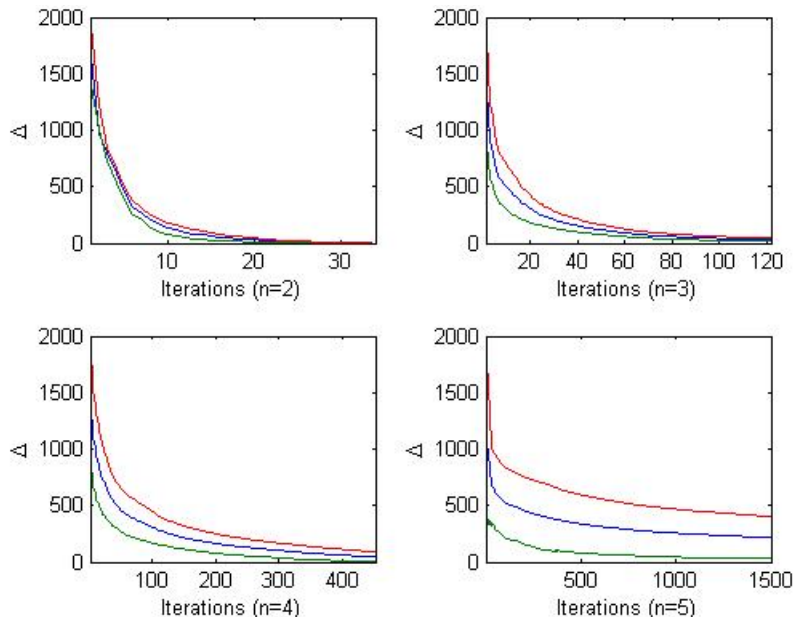


FIGURE 3.9 – Moyennes de l'écart d'approximation et intervalles de confiance à 95% sur les itérations

### 3.6.3 Remarques empiriques

Lorsqu'un seuil assez faible est fixé sur l'erreur d'approximation ( $\Delta$ ), comme l'illustre la figure 3.10, à mesure qu'ils sont divisés, certains simplexes peuvent devenir très plats. Leurs sommets peuvent donc être alignés (linéairement dépendants), la matrice  $\begin{pmatrix} X \\ e^T \end{pmatrix}$  peut devenir singulière (non inversible). Pour pallier ce problème, dans l'algorithme de division de simplexes, on peut fixer un seuil sur le volume relatif pour qu'un simplexe soit actif (à diviser). Ce seuil doit être déterminé empiriquement.

Il se peut également que sur un simplexe, le point de division soit très proche d'un sommet, ce qui peut aussi entraîner des difficultés numériques. Celles-ci peuvent être contournées en déplaçant légèrement le point de division vers le barycentre par exemple.

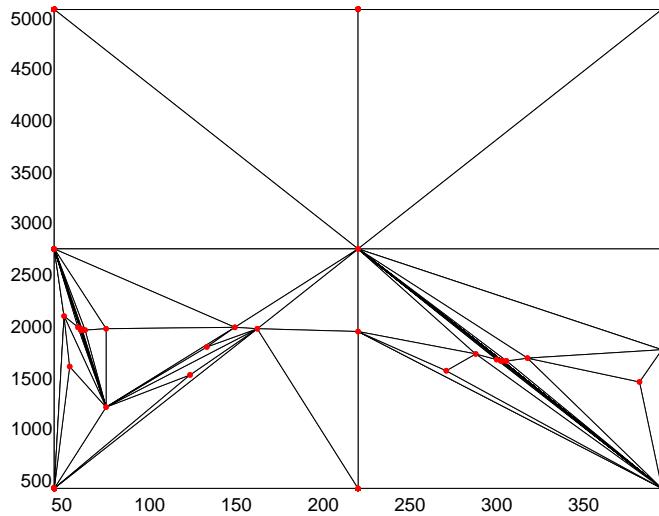


FIGURE 3.10 – Exemples de cas où les simplexes sont trop plats

### 3.7 Conclusion

Nous avons présenté dans ce chapitre une nouvelle approche pour approximer la fonction de valeur de la programmation dynamique stochastique. Le raffinement de l'approximation est implicitement guidé par l'incurvation de la fonction de valeur.

L'approche est basée sur la décomposition initiale de l'espace des stocks, supposé être un hyperrectangle, en simplexes. Ensuite, ces simplexes existants sont divisés de manière itérative si on veut raffiner la grille. Le procédé est répété jusqu'à l'atteinte d'un critère d'arrêt préalablement fixé, ce qui permet un compromis entre le temps de calcul et la précision souhaitée. Cependant, la complexité de la décomposition d'un hyperrectangle restreint les applications à moins de dix réservoirs.

Dans le chapitre qui suit, nous discutons d'une approche de programmation dynamique stochastique approximative basée sur la méthodologie présentée dans ce chapitre en insistant sur le processus information-décision. De plus, nous présentons des formules analytiques de calcul de l'espérance de la fonction de valeur dans un contexte où le système est situé dans un même bassin hydrographique.



# Chapitre 4

## Méthode de programmation dynamique approximative

### 4.1 Introduction

Le chapitre précédent a présenté une méthode d'approximation de la fonction de valeur pour une récurrence de l'algorithme de programmation dynamique stochastique. Ce chapitre traite d'une approche de *programmation dynamique stochastique approximative* complète (sur tout l'horizon de planification). L'approximation de la fonction de valeur est basée sur la méthodologie du partitionnement de l'espace des états en simplexes présentée dans le chapitre précédent. En plus d'hypothèses de concavité (proposition 3.4 du chapitre 3), nous utiliserons des hypothèses additionnelles sur les apports naturels qui nous permettront de proposer des formes analytiques pour l'espérance de la fonction de valeur.

La section 4.2 réexamine la modélisation d'un système de réservoirs et ses opérations avec une attention particulière au processus information-décision. Ceci établit les bases pour une récurrence de la PDS que nous cherchons à opérationnaliser. La section 4.3 fait une revue sommaire de quelques méthodes de programmation dynamique approximative. La section 4.4 aborde le calcul de l'espérance qui contribue en partie à la complexité de la PDS. Ce calcul peut être simplifié en exploitant une structure uni-bassin souvent partagée par les réservoirs. La section 4.5 traite de l'optimisation. Un programme linéaire généralisé est utilisé pour évaluer la fonction de valeur. Des expériences numériques réalisées ainsi que leurs résultats sont documentés dans la section 4.6.

## 4.2 Structure et contrôle d'un système de réservoirs

Rappelons qu'un système hydroélectrique est généralement composé de plusieurs réservoirs et de centrales de production, chacune pouvant être associée à un réservoir ou être au fil de l'eau. Les décisions opérationnelles portent d'une part sur le turbinage d'eau pour la production d'énergie et d'autre part sur le délestage (déversement) en cas de surplus d'eau.

Supposons que les décisions de turbinage soient prises avant d'observer les apports naturels. L'information disponible pour ces décisions sont donc les niveaux initiaux des réservoirs, et éventuellement l'observation antérieure d'une variable hydrologique pouvant être utile pour la prévision des apports naturels futurs. Le délestage se produit au contraire en temps continu, et est donc conditionné à la fois sur le turbinage et sur les apports naturels.

Le système étudié dans ce chapitre est le même que celui illustré dans la figure 1.2 du chapitre 1. Chaque nœud de ce système représente un réservoir ou une centrale au fil de l'eau ou un groupe réservoir-centrale et comprend deux réseaux correspondant au turbinage et au délestage.

Nous distinguons les types de nœuds via les ensembles d'indices suivants : (i)  $I_F$  : l'ensemble des nœuds avec centrale seule ; (ii)  $I_R$  : l'ensemble des nœuds avec réservoir (seul ou groupe centrale-réservoir), avec  $|I_R| = n$  ; (iii)  $I_C$  : l'ensemble des nœuds avec centrale, avec  $|I_C| = m$  et (iv)  $I_{RC} = I_R \cap I_C$  : l'ensemble des nœuds avec groupe centrale-réservoir.

Les apports naturels aux nœuds du réseau à la période  $t$ , notés  $Q_t$ , sont modélisés par un processus non stationnaire, stochastiquement dépendant d'une variable hydrologique scalaire notée  $\tilde{\epsilon}_t$ , qui forme un processus stochastique autorégressif d'ordre 1. On a donc :

$$\tilde{\epsilon}_t = \alpha \tilde{\epsilon}_{t-1} + e_t$$

où  $e_t$  est un processus « bruit blanc » (suite de variables indépendantes et identiquement distribuées) d'espérance nulle et de variance constante  $\sigma_e^2$ .

Pour tout nœud  $i \in I_{RC}$ , on note par  $f_{it}(u_{it}, v_{i,t-1}, v_{it}, \tilde{\epsilon}_{t-1})$  sa fonction de production à la période  $t$ . Cette fonction dépend du débit turbiné,  $u_{it}$ , de l'effet de hauteur de chute, pris en compte en considérant les niveaux des stocks en début et en fin de période



respectivement  $v_{i,t-1}$  et  $v_{it}$ , et de la variable hydrologique à la période précédente  $\tilde{\epsilon}_{t-1}$ . De même, pour tout nœud  $i \in I_F$ , la fonction de production dépend uniquement du flux d'eau turbiné  $u_{it}$ , puisqu'il n'y a pas de variation de stock, et est notée  $f_{it}(u_{it})$ .

Ainsi, à chaque période, l'état du système évolue suivant les équations de stocks suivantes :

$$v_t = v_{t-1} - B_{I_R}u_t - C_{I_R}y_t + Q_{I_Rt}$$

et les équations de conservation de l'eau ci-après :

$$Q_{I_Ft} - B_{I_F}u_t - C_{I_F}y_t = 0$$

où pour toute matrice  $A \in R^{p \times r}$ , tout vecteur  $a \in R^p$  et tout sous-ensemble  $I \subset \{1, \dots, p\}$ ,  $A_I(a_I)$  représente la sous-matrice (le sous-vecteur) obtenue en sélectionnant les lignes de  $A$  (les éléments de  $a$ ) indicées dans  $I$ .

La dernière équation traduit le fait que l'eau qui entre dans une centrale au fil de l'eau à la période  $t$  sort à la même période, puisqu'elle ne peut pas être stockée.

Les modèles d'optimisation cherchent généralement à maximiser l'espérance de la production d'énergie sur un horizon de planification. Dans ce contexte, cette optimisation peut être formulée comme :

$$\text{Max}_{u_t, y_t} \sum_{t=1}^T E_{Q_t | \tilde{\epsilon}_{t-1}} \left[ \sum_{i \in I_{RC}} f_{it}(u_{it}, v_{i,t-1}, v_{it}, \tilde{\epsilon}_{i,t-1}) + \sum_{i \in I_F} f_{it}(u_{it}) \right]$$

Sous les contraintes, pour  $1 \leq t \leq T$  :

$$v_t = v_{t-1} - B_{I_R}u_t - C_{I_R}y_t + Q_{I_Rt}$$

$$Q_{I_Ft} - B_{I_F}u_t - C_{I_F}y_t = 0$$

$$\underline{v}_t \leq v_t \leq \bar{v}_t$$

$$\underline{u}_t \leq u_t \leq \bar{u}_t$$

$$y_t \geq 0$$

$$u_t \text{ conditionné sur } v_{t-1} \text{ et } \epsilon_{t-1}$$

$$y_t \text{ et } v_t \text{ conditionnés sur } v_{t-1}, u_t \text{ et } Q_t$$

Sous les hypothèses précédentes, pour  $t = T + 1, T, \dots, 1$ , la récurrence de la PDS

s'écrit :

$$V_{t-1}(v_{t-1}, \epsilon_{t-1}) = \text{Max}_{u_t} \left\{ \sum_{i \in I_{RC}} \bar{f}_{it}(u_{it}, v_{i,t-1}, \epsilon_{i,t-1}) + \sum_{i \in I_F} f_{it}(u_{it}) + E_{Q_t, \tilde{\epsilon}_t | \epsilon_{t-1}} \left[ \text{Max}_{v_t, y_t} V_t(v_t, \tilde{\epsilon}_t) \right] \right\}$$

$$\text{S.l.c. } v_t = v_{t-1} - B_{I_R} u_t - C_{I_R} y_t + Q_{I_R t}$$

$$Q_{I_F t} - B_{I_F} u_t - C_{I_F} y_t = 0$$

$$\underline{v}_t \leq v_t \leq \bar{v}_t$$

$$\underline{u}_t \leq u_t \leq \bar{u}_t$$

$$y_t \geq 0$$

$$u_t \text{ conditionné sur } v_{t-1} \text{ et } \epsilon_{t-1}$$

$$y_t \text{ et } v_t \text{ conditionnés sur } v_{t-1}, u_t \text{ et } Q_t$$

où  $\bar{f}_{it}(u_{it}, v_{i,t-1}, \epsilon_{i,t-1}) = E_{Q_t | \tilde{\epsilon}_{t-1}} [f_{it}(u_{it}, v_{i,t-1}, v_{it}, \tilde{\epsilon}_{i,t-1})], \forall i \in I_{RC}$ .

Pour chaque période  $t$ , ce programme dynamique consiste à (i) calculer la politique optimale de délestage  $y_t^*$ , (ii) calculer une espérance et (iii) calculer la politique optimale des écoulements planifiés  $u_t^*$ .

Posons  $x_t = v_{t-1} - B_{I_R} u_t$ .  $x_t$  représente le niveau des réservoirs après les décisions  $u_t$  et avant la réalisation des apports naturels. Cette récurrence peut être explicitée en deux étapes : une étape recours « *ex post* » et une étape « *ex ante* » consistant à calculer la politique optimale des écoulements planifiés  $u_t^*$ .

Tout d'abord, les niveaux des réservoirs après turbinage sont connus. Les apports naturels  $Q_t$  et la variable hydrologique à la période précédente  $\tilde{\epsilon}_{t-1}$  sont observés. En cas de surplus d'eau, les décisions de délestage sont prises afin de maximiser la valeur (en unités d'énergie) de l'eau restante  $V_t$ . Les décisions optimales *ex post* sont donc une solution du problème suivant :

$$F_t(x_t, Q_t, \epsilon_t) = \text{Max}_{v_t, y_t} V_t(v_t, \epsilon_t) \quad (4.1)$$

$$\text{S.l.c. } v_t = x_t - C_{I_R} y_t + Q_{I_R t} \quad (4.2)$$

$$Q_{I_F t} - B_{I_F} u_t - C_{I_F} y_t = 0 \quad (4.3)$$

$$\underline{v}_t \leq v_t \leq \bar{v}_t \quad (4.4)$$

$$y_t \geq 0 \quad (4.5)$$

Ensuite, les niveaux des réservoirs au début de la période  $t$ ,  $v_{t-1}$ , sont observés et les décisions de turbinage sont prises afin de maximiser la production immédiate d'énergie et la valeur future espérée de l'eau. Les décisions ex ante sont donc une solution de :

$$V_{t-1}(v_{t-1}, \epsilon_{t-1}) = \text{Max}_{u_t, x_t} \left\{ \sum_{i \in I_{RC}} \bar{f}_{it}(u_{it}, v_{i,t-1}, \epsilon_{i,t-1}) + \sum_{i \in I_F} f_{it}(u_{it}) + E_{Q_t, \tilde{\epsilon}_t | \epsilon_{t-1}} [F_t(x_t, Q_t, \tilde{\epsilon}_t)] \right\}$$

$$\text{S.l.c. } x_t = v_{t-1} - B_{I_R} u_t$$

$$B_{I_F} u_t = 0$$

$$x_t \geq \underline{v}_t$$

$$\underline{u}_t \leq u_t \leq \bar{u}_t$$

La politique optimale de l'étape ex post a pour domaine un très vaste espace. Son calcul exact est donc onéreux. Il est cependant possible de formuler comme règle simple une politique de délestage cohérente avec la notion de valeur de l'eau croissante.

**Proposition 4.1.** *Si tous les nœuds ont une issue de délestage, la fonction  $V_{t-1}(v_{t-1}, \epsilon_{t-1})$  est non-décroissante en  $v_{t-1}$ .*

*Démonstration.* Le réseau de délestage est une arborescence. Supposons, sans perte de généralité, que les nœuds soient numérotés selon un tri topologique croissant. Alors la matrice d'incidence  $C$  de ce réseau est triangulaire inférieure. Comme chaque nœud a une issue de délestage, les éléments diagonaux de  $C$  sont tous égaux à 1. Par suite,  $C$  est non singulière, et  $C^{-1} \in \{0, 1\}^{p \times p}$  (Bazaraa *et al.*, 1990), où  $p$  est le nombre de nœuds du réseau.

Montrons que pour tout  $dv \geq 0$ ,  $V_{t-1}(v_{t-1} + dv, \epsilon_{t-1}) \geq V_{t-1}(v_{t-1}, \epsilon_{t-1})$ . Pour simplifier la notation, la variable hydrologique  $\epsilon_{t-1}$ , étant fixée, sera omise. Définissons

$$\phi_t(v_{t-1}, u_t, Q_t, y_t) = \sum_{i \in I_{RC}} f_{it}(u_{it}, v_{i,t-1}, v_{it}, \tilde{\epsilon}_{t-1}) + \sum_{i \in I_F} f_{it}(u_{it}) + V_t(v_{t-1} - Bu_t - Cy_t + Q_t, \epsilon_t)$$

La fonction  $\phi_t$  mesure la valeur de l'eau pour des niveaux de stocks  $v_{t-1}$ , des décisions de turbinage  $u_t$ , des apports naturels observés  $Q_t$  et des décisions de délestage  $y_t$  donnés. Étant donné un niveau de stock initial  $v_{t-1}$ , à la période  $t$ , la politique optimale consiste en une règle  $u_t^*(v_{t-1})$  suivie d'une règle  $y_t^*(v_{t-1} - Bu_t + Q_t)$ . Par suite,

$$V_{t-1}(v_{t-1}) = E_{Q_t} [\phi_t(v_{t-1}, u_t^*(v_{t-1}), Q_t, y_t^*(v_{t-1} - Bu_t^*(v_{t-1}) + Q_t))]$$

Supposons que le stock initial augmente de  $v_{t-1}$  à  $v_{t-1} + dv$  ( $dv \geq 0$ ). Considérons la nouvelle politique :

$$\hat{u}_t(\hat{v}_{t-1}) = u_t^*(v_{t-1})$$

$$\hat{y}_t(\hat{v}_{t-1} - B\hat{u}_t(\hat{v}_{t-1}) + Q_t) = y_t^*(v_{t-1} - Bu_t^*(v_{t-1}) + Q_t) + dy$$

avec  $Cdy = dv$ .

Cette politique est réalisable. En effet (i) les écoulements planifiés restent inchangés et satisfont donc leurs bornes, (ii) les stocks finaux restent inchangés, (iii) les délestages augmentent de  $dy = C^{-1}v$  et restent donc non négatifs.

Par ailleurs, on vérifie que

$$\begin{aligned} \phi_t(v_{t-1}, \hat{u}_t(\hat{v}_{t-1}), Q_t, \hat{y}_t(v_{t-1} + dv - B\hat{u}_t(\hat{v}_{t-1}) + Q_t)) = \\ \phi_t(v_{t-1}, u_t^*(v_{t-1}), Q_t, y_t^*(v_{t-1} - Bu_t^*(v_{t-1}) + Q_t)) \end{aligned}$$

Comme la nouvelle politique est réalisable, mais pas nécessairement optimale pour le niveau de stock initial  $v_{t-1} + dv$ , on a :

$$\begin{aligned} V_{t-1}(v_{t-1} + dv) &\geq E_{Q_t} [\phi_t(v_{t-1}, \hat{u}_t(\hat{v}_{t-1}), Q_t, \hat{y}_t(v_{t-1} + dv - B\hat{u}_t(\hat{v}_{t-1}) + Q_t))] \\ &= E_{Q_t} [\phi_t(v_{t-1}, u_t^*(v_{t-1}), Q_t, y_t^*(v_{t-1} - Bu_t^*(v_{t-1}) + Q_t))] \\ &= V_{t-1}(v_{t-1}) \end{aligned}$$

□

Étant donné que  $V_{t-1}$  est non décroissante, les décisions optimales de délestage pour (4.1-4.5) peuvent être trouvées par la règle simple suivante (Lamond et Lang, 2007) :

**Proposition 4.2.** *Soit  $P_i$  l'ensemble des prédécesseurs du nœud  $i$  dans le réseau de délestage. Sous les hypothèses de la proposition 4.1, si  $x_t \geq \underline{v}_t$ , il existe une politique unique Pareto-minimale donnée récursivement par :*

$$y_{it}^*(x_t + Q_t) = \max \{0, x_{it} + Q_{it} + \sum_{j \in P_i} y_{jt}^*(x_t + Q_t) - \bar{v}_{it}\}$$

avec les conventions  $x_{it} = 0 \forall i \in I_F$  et  $\sum_{j \in \emptyset(\cdot)_j} = 0$ . Cette politique est optimale pour le problème de délestage. Le stock final résultant est :

$$v_{it}^*(x_t + Q_t) = \min \{\bar{v}_{it}, x_{it} + Q_{it} + \sum_{j \in P_i} y_{jt}^*(x_t + Q_t)\} \quad (4.6)$$

Notons que les stocks finaux optimaux satisfont les conditions  $v_t^*(x_t + Q_t) \geq \underline{v}_t$ . En cas de surplus d'eau (délestage), les bornes supérieures sur les stocks sont saturées, i.e.  $(\bar{v}_t - v_t^*)^T y_t^* = 0$ .

Avec cette politique de délestage, le programme dynamique stochastique se simplifie en :

$$V_{t-1}(v_{t-1}, \epsilon_{t-1}) = \text{Max}_{u_t, x_t} \left\{ \sum_{i \in I_{RC}} \bar{f}_{it}(u_{it}, v_{i,t-1}, \epsilon_{i,t-1}) + \sum_{i \in I_F} f_{it}(u_{it}) + E_{Q_t, \tilde{\epsilon}_t | \epsilon_{t-1}} [V_t(v_t^*(x_t + Q_{I_{Rt}}), \tilde{\epsilon}_t)] \right\} \quad (4.7)$$

$$\text{S.l.c. } x_t = v_{t-1} - B_{I_R} u_t \quad (4.8)$$

$$B_{I_F} u_t = 0 \quad (4.9)$$

$$x_t \geq \underline{v}_t \quad (4.10)$$

$$\underline{u}_t \leq u_t \leq \bar{u}_t \quad (4.11)$$

### 4.3 Programmation dynamique approximative

Pour obtenir une précision acceptable, la fonction de valeur doit être évaluée en un nombre suffisant de points. Une difficulté commune est que les efforts de calcul induits par le nombre de points de grille augmentent exponentiellement avec la dimension de l'espace des états (Rust, 1996, Grüne et Semmler, 2004). Bellman (1957), l'auteur de la PD, fut le premier à approximer la fonction de valeur par un polynôme.

Plusieurs modèles numériques ont été proposés pour approximer la fonction de valeur tels que les polynômes de Chebyshev (e.g. Rust, 1996), des approximations par des splines (e.g. Johnson *et al.*, 1993), des approximations par différences finies (e.g. Candler, 2001). Cependant, ces approches exigent que la fonction de valeur soit lisse<sup>1</sup>. Grüne et Semmler (2004) présentent plus de détails sur ces techniques.

Développée par Pereira et Pinto (1985), la *programmation dynamique stochastique duale* (PDSO) approxime la fonction de valeur par des fonctions linéaires par morceaux. Celles-ci, équivalentes à des *coupes de Benders*, sont construites à partir d'information fournie par le dual du programme original. La PDSO a été implantée par Tilmant *et al.* (2008) pour évaluer la valeur marginale de l'eau et sa distribution statistique à différents points dans l'espace et le temps pour un système de sept réservoirs.

Foufoula-Georgiou et Kitanidis (1988) ont développé un algorithme basé sur l'interpolation d'Hermite appelé *programmation dynamique par gradient* pour un modèle multi-réservoirs. La fonction de valeur est approximée localement par le polynôme de plus

---

1. Une fonction est lisse si toutes ses dérivées partielles existent et sont continues.

petit ordre qui préserve les valeurs de la fonction et son gradient par rapport aux variables d'état à chaque point de discrétisation.

En procédant par induction à rebours, à chaque période, une politique est déterminée pour chaque niveau des réservoirs par la méthode de Newton. À la période  $t$ , si aucune des contraintes d'inégalité n'est active, à la période  $t - 1$ , un programme d'optimisation sans contrainte est résolu. Si certaines contraintes sont actives à la période  $t$ , à la période précédente le programme est résolu en considérant uniquement ces dernières. Cette stratégie permet de calculer à chaque itération le gradient de la fonction de valeur.

Gal (1989) propose une autre méthode appelée *itération sur les paramètres*. À chaque période, la fonction de valeur est approximée par une fonction choisie parmi une famille admissible dépendant d'un ensemble de paramètres inconnus, par exemple, les coefficients d'une fonction linéaire.

Un vecteur de politique initiale « raisonnable » est utilisé et des trajectoires des niveaux des réservoirs sont simulées à partir de certaines de ces politiques. Une des fonctions admissibles est alors choisie pour approximer, par induction à rebours, la fonction de valeur à partir de ces niveaux des réservoirs. À chaque itération, les valeurs des paramètres sont améliorées et un nouveau vecteur de politique est déterminé.

Le principal désavantage de cette approche est qu'elle exige une bonne connaissance du système de réservoirs afin de pouvoir choisir la famille des fonctions admissibles.

Une technique plus récente, appelée *programmation neuro-dynamique* (PND), consistant à approximer la fonction de valeur par des réseaux de neurones (RN), a été proposée par Bertsekas et Tsitsiklis (1995). Le modèle est résolu pour un certain nombre de points de discrétisation, puis le réseau est *entraîné* afin de simuler la politique de gestion en des points différents.

La PND a été appliquée à des modèles multi-réservoirs par Cervellera *et al.* (2006) et Castelletti *et al.* (2007). Dans le premier cas, un modèle à dix réservoirs a été résolu sur un horizon de trois périodes uniquement.

Une autre approche inspirée de l'intelligence artificielle et des sciences cognitives est l'*apprentissage par renforcement* où la fonction de valeur est construite par simulation. À chaque itération, cette fonction est calculée pour un seul état, obtenu par une équation de transition, et un ensemble de décisions choisies aléatoirement, ce qui permet de réduire l'espace des états et des décisions (Lee et Labadie, 2007). Des applications aux

modèles de réservoirs sont présentées dans les travaux de Lee et Labadie (2007) et Castelletti *et al.* (2010).

Finalement, mentionnons des méthodes basées sur des métaheuristiques telles que les algorithmes génétiques. Des exemples d'application peuvent être trouvés dans Ahmed et Sarma (2005), Jothiprakash et Shanthi (2006), et Chen et Chang (2007).

L'estimation locale de l'erreur d'approximation constitue la principale différence entre notre approche et celles décrites plus haut.

## 4.4 Calcul de l'espérance de la fonction de valeur

Afin de limiter les évaluations de la fonction  $V_{t-t}$ , nous utiliserons la méthodologie discutée dans le chapitre précédent. Rappelons qu'à la période  $t-1$ , l'espace des stocks finaux est un hyperrectangle  $H_{t-1} = \{v_{t-1} \in \mathbf{R}^n \mid \underline{v}_{t-1} \leq v_{t-1} \leq \bar{v}_{t-1}\}$  partitionné en simplexes de pleine dimension.

Le problème (4.7-4.11) a deux sources importantes de complexité : (i) un calcul d'espérance et (ii) une optimisation. L'objectif de cette section est le calcul de l'espérance  $E_{Q_t, \tilde{\epsilon}_t | \epsilon_{t-1}} [V_t(v_t^*(x_t + Q_{I_{Rt}}), \tilde{\epsilon}_t)]$  pour tous les états  $(x_t, \epsilon_{t-1})$ . Sans perte de généralité, nous supposons que la variable hydrologique évolue suivant une loi de probabilité de transition connue  $P(\tilde{\epsilon}_t = \epsilon_t \mid \tilde{\epsilon}_{t-1} = \epsilon_{t-1})$ . Ceci revient à considérer que la variable hydrologique est Markovienne, une approche très utilisée dans les problèmes de réservoirs (e.g. Tejada-Guibert *et al.*, 1995 et Goor *et al.*, 2010). Nous avons donc une corrélation sérielle représentée par un processus Markovien discret. L'espérance est alors donnée par :

$$\begin{aligned} E_{Q_t, \tilde{\epsilon}_t | \epsilon_{t-1}} [V_t(v_t^*(x_t + Q_{I_{Rt}}), \tilde{\epsilon}_t)] &= E_{\tilde{\epsilon}_t | \epsilon_{t-1}} E_{Q_t | \epsilon_{t-1}} [V_t(v_t^*(x_t + Q_{I_{Rt}}), \tilde{\epsilon}_t)] \\ &= \sum_{\epsilon_t} P(\tilde{\epsilon}_t = \epsilon_t \mid \tilde{\epsilon}_{t-1} = \epsilon_{t-1}) \\ &\quad E_{Q_t | \epsilon_{t-1}} [V_t(v_t^*(x_t + Q_{I_{Rt}}), \epsilon_t)] \end{aligned}$$

Prendre l'espérance revient donc à calculer :

$$J_t(x_t, \epsilon_{t-1}) = E_{Q_t | \epsilon_{t-1}} [V_t(v_t^*(x_t + Q_{I_{Rt}}), \epsilon_t)]$$

pour un ensemble d'états discrets  $(x_t, \epsilon_{t-1})$ . Nous avons discuté de la discrétisation des stocks  $(x_t)$  dans le chapitre précédent. Nous exposerons des stratégies de discrétisation de la variable hydrologique dans le chapitre suivant.

Le calcul de l'espérance peut encore être simplifié si de plus, on a une corrélation spatiale parfaite (entre les apports naturels). Cette hypothèse « uni-bassin » est approximative-

ment vérifiée dans plusieurs situations pratiques où les réservoirs sont alimentés par un même bassin hydrographique. D'un point de vue pratique, la dimension du support des apports naturels passe de  $n$  à 1. Cette hypothèse est souvent utilisée par les hydrologues dans le cas d'horizon de planification divisé en pas de temps mensuel ou hebdomadaire (Turgeon, 2007).

La corrélation spatiale parfaite revient à considérer une variable aléatoire scalaire  $\tilde{\tau}_t \geq 0$  telle que

$$Q_t = q_t \tilde{\tau}_t$$

où  $q_t \in \mathbf{R}^n$  représente les contributions relatives des apports naturels aux réservoirs, et  $\tilde{\tau}_t \geq 0$ .

On pourrait utiliser l'intégration numérique pour calculer la fonction  $J_t(x_t, \epsilon_{t-1})$  si la distribution de probabilité de  $\tilde{\tau}_t$  était connue. Toutefois, la partition de l'espace des stocks en simplexes nous permettra d'obtenir des formes analytiques pour l'espérance. Dans un premier temps, nous analyserons la trajectoire des stocks finaux dans cet espace partitionné. L'approximation de  $V_t$  dans cet espace partitionné est notée  $\hat{V}_t$ . Ensuite, nous aborderons l'évaluation de  $\hat{V}_t$  sur cette trajectoire.

Pour tout simplexe  $S(X)$ ,  $\hat{V}_t$  est évalué aux différents sommets  $x^i$ . Ces évaluations sont notées par  $z_i = \hat{V}_t(x^i, \epsilon_t)$ ,  $i = 1, \dots, n + 1$ . Pour tout autre point  $x \in S(X)$ ,  $\hat{V}_t$  est interpolée sur  $S(X)$ . Rappelons que cette interpolation est uniquement déterminée par la fonction affine :

$$\hat{V}_t(x, \epsilon_t) = z^T \begin{pmatrix} X \\ e^T \end{pmatrix}^{-1} \begin{pmatrix} x \\ 1 \end{pmatrix}, \text{ où } z^T = (z_1, \dots, z_{n+1}).$$

Sous l'hypothèse uni-bassin, la trajectoire des stocks finaux et du délestage sont des lignes unidimensionnelles dans  $\mathbf{R}^n$ . De l'équation (4.6), nous voyons que dans le cas des stocks finaux, la trajectoire  $\hat{v}_t(\tau) \triangleq v_t^*(x_t + q_t \tau)$  est linéaire jusqu'à ce qu'une borne supérieure des réservoirs soit atteinte. Nous appellerons *coin* un point où une nouvelle borne devient saturée. À chaque nouveau coin, la trajectoire change de direction, la nouvelle borne restant saturée jusqu'à la fin de la trajectoire  $\bar{v}_t$ . La trajectoire  $\hat{v}_t(\tau)$  est illustrée dans la figure 4.1.

En plus des coins, la trajectoire  $\hat{v}_t(\tau)$  traverse les simplexes à des points de traversée intermédiaires (voir la section 3.5 pour un rappel). Nous pouvons définir un ensemble de *nœuds*  $\Theta = \{\theta_i \mid i = 1, \dots, w\}$  du paramètre  $\tilde{\tau}_t$  tels que  $\hat{v}_t(\theta_i)$  soit un coin ou un point de traversée.



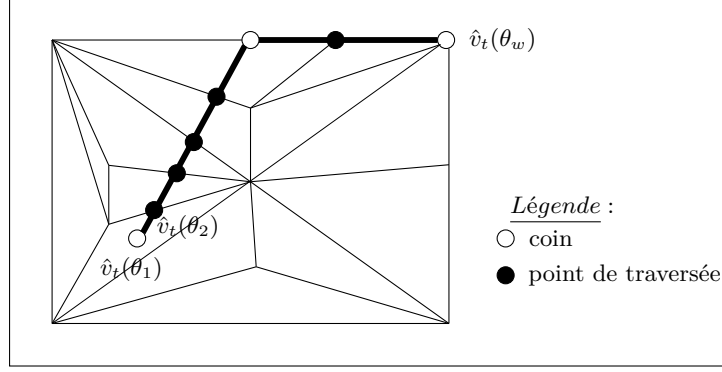


FIGURE 4.1 – Exemple de trajectoire des stocks finaux

La fonction  $\hat{V}_t$  est évaluée à chaque nœud  $\theta_i, i = 1, \dots, w$ . Soit  $h_t(\theta_i, \epsilon_t) = \hat{V}_t(\hat{v}_t(\theta_i), \epsilon_t)$ . Les pentes de la fonction  $h_t$  changent aux nœuds dans  $\Theta$ . Ainsi,  $h_t$  est affine par morceaux. Pour chaque segment  $(\theta_{i-1}, \theta_i)$ , la pente est constante et est donnée par :

$$\alpha_i = \frac{\omega_i - \omega_{i-1}}{\theta_i - \theta_{i-1}}, \quad i = 2, \dots, w + 1$$

où  $\omega_i = \hat{V}_t(\hat{v}_t(\theta_i), \epsilon_t)$ . En particulier,  $\alpha_{w+1} = 0$ . La fonction  $h_t$  est constante pour  $\tau \geq \theta_w$ , les stocks atteignant leur niveau maximal. Nous avons donc

$$h_t(\theta_i, \epsilon_t) = \begin{cases} \omega_i & i = 2, \dots, w \\ \omega_w & i = w + 1 \end{cases}$$

Alors, la fonction  $h_t$  est de la forme :

$$h_t(\tau, \epsilon_t) = \begin{cases} \omega_{i-1} + \alpha_i(\tau - \theta_{i-1}) & \theta_{i-1} \leq \tau \leq \theta_i, \quad i = 2, \dots, w \\ \omega_w & \tau \geq \theta_w \end{cases}$$

#### 4.4.1 Forme analytique

La fonction  $J_t(x_t, \epsilon_{t-1})$  est calculée sur chaque segment  $(\theta_{i-1}, \theta_i), i = 2, \dots, w + 1$ . Soit  $F_t(\tau)$  la fonction de répartition de  $\tilde{\tau}_t$  conditionnée sur la valeur  $\epsilon_{t-1}$ . L'espérance conditionnelle est donc donnée par :

$$J_t(x_t, \epsilon_{t-1}) = \sum_{i=2}^{w+1} \int_{\theta_{i-1}}^{\theta_i} h_t(\tau, \epsilon_t) dF_t(\tau)$$

où

$$\begin{aligned}
\sum_{i=2}^{w+1} \int_{\theta_{i-1}}^{\theta_i} h_t(\tau, \epsilon_t) dF_t(\tau) &= \sum_{i=2}^{w+1} \int_{\theta_{i-1}}^{\theta_i} (\omega_{i-1} + \alpha_i(\tau - \theta_{i-1})) dF_t(\tau) \\
&= \sum_{i=2}^{w+1} \alpha_i E[\tilde{\tau}_t | \epsilon_{t-1}]_{\theta_{i-1}}^{\theta_i} + \sum_{i=2}^{w+1} \gamma_{i-1} [F_t(\theta_i) - F_t(\theta_{i-1})] \\
&= \sum_{i=2}^{w+1} \alpha_i E[\tilde{\tau}_t | \epsilon_{t-1}]_{\theta_{i-1}}^{\theta_i} + \sum_{i=2}^w \gamma_{i-1} [F_t(\theta_i) - F_t(\theta_{i-1})] + \omega_w [1 - F_t(\theta_w)]
\end{aligned}$$

avec  $\gamma_i = \omega_i - \alpha_{i+1}\theta_i$  et  $\theta_{w+1} = +\infty$ .

Nous détaillons les cas des distributions normale et log-normale.

#### 4.4.2 Cas de la distribution normale tronquée

Supposons que  $\tilde{\tau}_t$  soit une variable aléatoire normale de moyenne  $\mu(\epsilon_{t-1})$  et d'écart type  $\sigma(\epsilon_{t-1})$ . On a :

$$\begin{aligned}
F_t(\tau) &= P(\tilde{\tau}_t \leq \tau) = \Phi\left(\frac{\tau - \mu(\epsilon_{t-1})}{\sigma(\epsilon_{t-1})}\right) \\
f_t(\tau) &= F_t'(\tau) = \frac{1}{\sigma(\epsilon_{t-1})} \phi\left(\frac{\tau - \mu(\epsilon_{t-1})}{\sigma(\epsilon_{t-1})}\right)
\end{aligned}$$

où  $\phi(s) = \frac{1}{\sqrt{2\pi}} e^{-\frac{s^2}{2}}$  et  $\Phi(s) = \int_{-\infty}^s \phi(t) dt$  sont respectivement la fonction de densité et la fonction de répartition de la distribution normale centrée réduite. Soit  $G(\tau) = \mu(\epsilon_{t-1})F_t(\tau) - (\sigma(\epsilon_{t-1}))^2 f_t(\tau)$ . La loi normale centrée réduite a la propriété :  $\phi'(s) = -s\phi(s)$ . Il s'ensuit que

$$\int \tau f_t(\tau) d\tau = G(\tau)$$

Par conséquent,

$$E[\tilde{\tau}_t | \epsilon_{t-1}]_{\theta_{i-1}}^{\theta_i} = G(\theta_i) - G(\theta_{i-1})$$

et

$$J_t(x_t, \epsilon_{t-1}) = \frac{1}{1 - F_t(0)} \sum_{i=2}^{w+1} \{\alpha_i [G(\theta_i) - G(\theta_{i-1})] + \gamma_{i-1} [F_t(\theta_i) - F_t(\theta_{i-1})]\}$$

#### 4.4.3 Cas de la distribution log-normale

$\tilde{\tau}_t$  est une variable aléatoire log-normale de paramètres  $\mu(\epsilon_{t-1})$  et  $\sigma(\epsilon_{t-1})$  si  $\ln \tilde{\tau}_t$  est une variable aléatoire normale de moyenne  $\mu(\epsilon_{t-1})$  et de variance  $(\sigma(\epsilon_{t-1}))^2$ . La fonction de densité de  $\tilde{\tau}_t$  est :  $f_t(\tau; \mu(\epsilon_{t-1}), \sigma(\epsilon_{t-1})) = \frac{1}{\sqrt{2\pi}\sigma(\epsilon_{t-1})\tau} e^{-\frac{(\ln \tau - \mu(\epsilon_{t-1}))^2}{2(\sigma(\epsilon_{t-1}))^2}}$  ; sa moyenne est  $\nu =$

$E[\tilde{\tau}_t] = e^{\mu(\epsilon_{t-1}) + \frac{(\sigma(\epsilon_{t-1}))^2}{2}}$ , et son  $k^{\text{ème}}$  moment par rapport à l'origine  $\nu_k = E[\tilde{\tau}_t^k | \epsilon_{t-1}] = e^{k\mu(\epsilon_{t-1}) + \frac{k^2(\sigma(\epsilon_{t-1}))^2}{2}}$ .

Comme précédemment,  $\Phi$  représente la fonction de répartition de la variable normale centrée réduite. Soit  $\rho(v) = \frac{\ln v - \mu - \sigma^2}{\sigma}$ . Nous concluons avec le résultat suivant :

**Proposition 4.3.** *Soient  $0 \leq a < b$ . Alors  $E[\tilde{\tau}_t]_a^b = \nu [\Phi(\rho(b)) - \Phi(\rho(a))]$ .*

*Démonstration.* On a

$$\begin{aligned} E[\tilde{\tau}_t]_a^b &= E[e^Z]_{\ln a}^{\ln b}, \text{ où } Z \sim N(\mu, \sigma^2) \\ &= \frac{1}{\sqrt{2\pi\sigma}} \int_{\ln a}^{\ln b} e^{-\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2 + z} dz. \end{aligned}$$

On a aussi :

$$\begin{aligned} -\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2 + z &= -\frac{1}{2\sigma^2}(z^2 - 2\mu z + \mu^2 - 2\sigma^2 z) \\ &= -\frac{1}{2\sigma^2}(z^2 - 2(\mu + \sigma^2)z + \mu^2) \\ &= -\frac{1}{2\sigma^2}[(z - (\mu + \sigma^2))^2 - (\mu + \sigma^2)^2 + \mu^2] \\ &= -\frac{(z - (\mu + \sigma^2))^2}{2\sigma^2} + \mu + \frac{\sigma^2}{2}, \end{aligned}$$

d'où

$$E[\tilde{\tau}_t]_a^b = \frac{\nu}{\sqrt{2\pi\sigma}} \int_{\ln a}^{\ln b} e^{-\frac{1}{2}\left(\frac{z-\mu-\sigma^2}{\sigma}\right)^2} dz.$$

Par le changement de variable  $t = \frac{z-\mu-\sigma^2}{\sigma}$ , on obtient

$$E[\tilde{\tau}_t]_a^b = \nu \int_{\rho(a)}^{\rho(b)} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt.$$

L'intégrande étant la densité de la loi normale centrée réduite, la proposition est prouvée.  $\square$

Finalement, observons que la fonction de répartition de  $\tilde{\tau}_t$  est :  $F_t(\tau) = \Phi(\rho(\tau) + \sigma(\epsilon_{t-1}))$ .

Alors

$$J_t(x_t, \epsilon_{t-1}) = \sum_{i=2}^{w+1} \{ \nu \alpha_i [\Phi(\rho(\theta_i)) - \Phi(\rho(\theta_{i-1}))] + \gamma_{i-1} [\Phi(\rho(\theta_i) + \sigma(\epsilon_{t-1})) - \Phi(\rho(\theta_{i-1}) + \sigma(\epsilon_{t-1}))] \}$$

## 4.5 Optimisation par la programmation linéaire généralisée

Cette section est une généralisation de la section 3.4.

Une fois l'espérance de  $V_t$  calculée, le problème (4.7-4.11) sera résolu pour trouver la politique optimale de turbinage  $u_t^*$  pour un ensemble discret de valeurs  $(v_{t-1}, \epsilon_{t-1})$ . À cette fin, nous aurons besoin d'évaluations des fonctions de production. Celles-ci sont très souvent fortement non concaves. Il s'ensuit alors que le problème (4.7-4.11) peut être non convexe. Il n'est donc pas garanti de trouver des optima globaux.

La non concavité peut être causée par plusieurs facteurs. La puissance des turbines varie de façon non linéaire avec les flux d'eau. En effet, un débit minimal est nécessaire pour mettre en marche les turbines. Celles-ci atteignent leur rendement maximal à un certain débit donné. Au-delà d'un certain seuil, des turbulences peuvent diminuer l'efficacité.

Deuxièmement, l'énergie totale produite dépend du nombre de turbines en opération. Dépendamment des objectifs de production, des mises hors tension peuvent avoir lieu. Finalement, et non pas les moindres, viennent les effets de hauteur de chute. Un effet de non linéarité additionnelle se produit quand les débits turbinés affectent la hauteur de chute, par exemple en faisant augmenter le niveau du *bief aval*.

La figure 4.2 montre une fonction de production hypothétique ainsi qu'une approximation concave affine. Les points de cassure correspondent à des changements de groupe. Pour un problème moyen terme, cette approximation est acceptable, car il y a un modèle de court terme qui répartit le débit optimal entre les différents groupes turbo-alternateurs (problèmes de chargement optimal). L'optimisation de la gestion à court terme est donc beaucoup plus fine que celle à moyen terme.

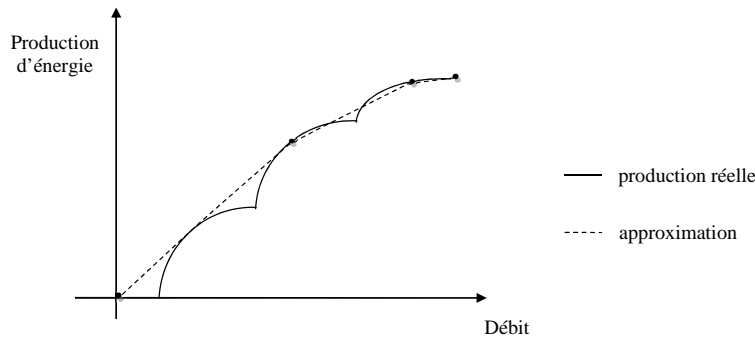


FIGURE 4.2 – Exemple d'une fonction de production et son approximation

Revenons maintenant à la résolution du problème (4.7-4.11). Étant donné que son domaine réalisable est un polyèdre convexe, en principe, ce problème peut être résolu à l'optimalité si la fonction de valeur espérée  $J_t$  est concave en  $x_t$  et les fonctions de production  $f_{it}$  sont concaves dans les débits  $u_{it}$ . Cependant, même dans ce cas convexe,

toutes les méthodes d'optimisation sous contraintes ne sont pas également appropriées. Le simple fait que le domaine réalisable soit un polyèdre entraîne que ni la politique optimale ni la fonction de valeur ne sont lisses.

La *programmation linéaire généralisée* (PLG, e.g. Shapiro, 1979) interpole la valeur de fonctions sur un ensemble discret de points. Si le problème original est convexe, une densification bien conçue de l'échantillon convergera ultimement vers le problème original. Dans le cas non convexe (e.g. ici si les fonctions de production ne sont pas concaves), la PLG produit une approximation convexe (« convexification ») du problème original. Par exemple, en *génération de colonnes*, le *problème maître* est sous forme de PLG.

Puisque le domaine réalisable est convexe, la PLG est une approximation appropriée du problème (4.7-4.11). Pour implanter la PLG, nous aurons besoin (i) d'une grille d'états  $\{\hat{x}_{jt} \mid j \in \Upsilon_t\}$  sur laquelle la fonction  $\hat{J}_t(x_t, \epsilon_{t-1}) = E_{Q_t, \tilde{\epsilon}_t | \epsilon_{t-1}} [\hat{V}_t(v_t^*(x_t + Q_{I_{Rt}}), \tilde{\epsilon}_t)]$  a été évaluée, et (ii) pour chaque centrale  $i$ , d'un échantillon de points  $\{\hat{u}_{ijt} \mid j \in \Gamma_{it}\}$  où la fonction de production a été évaluée aussi. De plus, nous distinguons les centrales associées à un réservoir ( $i \in I_{RC}$ ) de celles au fil de l'eau ( $i \in I_F$ ). Étant donné ces échantillons fixes, la PLG cherche une « meilleure » interpolation :

$$\hat{V}_{t-1}(v_{t-1}, \epsilon_{t-1}) = \text{Max}_{u_t, x_t, \lambda, \mu} \left\{ \sum_{i \in I_{RC}} \sum_{j \in \Gamma_{it}^1} \bar{f}_{it}(\hat{u}_{ijt}^1, v_{i,t-1}) \lambda_{ij}^1 + \sum_{i \in I_F} \sum_{j \in \Gamma_{it}^2} f_{it}(\hat{u}_{ijt}^2) \lambda_{ij}^2 + \sum_{j \in \Upsilon_t} \hat{J}_t(\hat{x}_{jt}, \epsilon_{t-1}) \mu_j \right\} \quad (4.12)$$

$$\text{S.l.c. } x_t = v_{t-1} - B_{I_R} u_t \quad (4.13)$$

$$B_{I_F} u_t = 0 \quad (4.14)$$

$$x_t = \sum_{j \in \Upsilon_t} \hat{x}_{jt} \mu_j \quad (4.15)$$

$$\sum_{j \in \Upsilon_t} \mu_j = 1 \quad (4.16)$$

$$x_t \geq \underline{v}_t \quad (4.17)$$

$$\underline{u}_t \leq u_t \leq \bar{u}_t \quad (4.18)$$

$$u_{it} = \sum_{j \in \Gamma_{it}^1} \hat{u}_{ijt}^1 \lambda_{ij}^1 \quad i \in I_{RC} \quad (4.19)$$

$$\sum_{j \in \Gamma_{it}^1} \lambda_{ij}^1 = 1 \quad i \in I_{RC} \quad (4.20)$$

$$u_{it} = \sum_{j \in \Gamma_{it}^2} \hat{u}_{ijt}^2 \lambda_{ij}^2 \quad i \in I_F \quad (4.21)$$

$$\sum_{j \in \Gamma_{it}^2} \lambda_{ij}^2 = 1 \quad i \in I_F \quad (4.22)$$

$$\lambda, \mu \geq 0 \quad (4.23)$$

où  $u_t = (u_t^1, u_t^2)^T$ ,  $\hat{u}_t = (\hat{u}_t^1, \hat{u}_t^2)^T$  et  $\lambda = (\lambda^1, \lambda^2)^T$ . (Pour un rappel sur la signification des contraintes, voir la section 3.4.)

Mais l'erreur d'approximation de la PLG est-elle à la hausse ou à la baisse? Ceci est difficilement prévisible en général. Toutefois, comme base de comparaison, nous avons une réponse claire dans le cas convexe.

**Proposition 4.4.** *Si les fonctions de production  $\bar{f}_{it}$  sont concaves en  $u_{it}$ , et si  $\hat{V}_t \leq V_t$ , alors la PLG (4.12-4.23) sous-estime la vraie production à la période  $t$ , et  $\hat{V}_{t-1} \leq V_{t-1}$ .*

*Démonstration.* Puisque les fonctions de production  $\bar{f}_t$  sont concaves en  $u_t$ , pour les mêmes débits, la valeur de la production évaluée comme la meilleure interpolation (combinaison convexe) sur les échantillons de points  $\{\hat{u}_{jt} \mid j \in \Gamma_t\}$  est une borne inférieure sur la vraie production. En outre, étant donné que  $\hat{V}_t \leq V_t$ , nous avons  $\hat{J}_t(x_t, \epsilon_{t-1}) \leq E_{Q_t, \tilde{\epsilon}_t | \epsilon_{t-1}} [V_t(v_t^*(x_t + Q_{I_R t}), \tilde{\epsilon}_t)] \forall x_t$  (par la propriété de linéarité de l'espérance).

Par conséquent, pour tout niveau post turbinage des réservoirs  $x_t$ , l'espérance évaluée comme la meilleure interpolation sur l'échantillon de point  $\{\hat{x}_{jt} \mid j \in \Upsilon_t\}$  est sous-estimée. Observons aussi que le polyèdre (4.13-4.23) (domaine réalisable de  $\hat{V}_{t-1}$ ) est inclus dans le polyèdre (4.8-4.11) (domaine réalisable de  $V_{t-1}$ ). Puisque l'objectif est maximisé, notre affirmation est prouvée.  $\square$

Finalement, nous devons noter qu'alors que la politique  $u_t^*(v_{t-1}, \epsilon_{t-1})$  est calculée exactement sur un ensemble fini de points de grille, elle peut être approximativement étendue sur le continuum de l'espace des états. En effet, supposons qu'un niveau initial de stock particulier  $v_{t-1}$  ne soit pas un point de grille, on peut trouver une politique approximative  $\hat{u}_t(v_{t-1}, \epsilon_{t-1})$  de la manière suivante.

Supposons que  $v_{t-1}$  appartienne au simplexe actif  $S(X)$  (voir la section 3.5), avec  $X = (v_{t-1}^1, \dots, v_{t-1}^{n+1})$ . Nous avons l'évaluation exacte de la fonction  $\hat{V}_{t-1}(v_{t-1}^i, \epsilon_{t-1})$  en chaque sommet  $v_{t-1}^i, i = 1, \dots, n+1$ , ainsi que la politique de turbinage correspondante  $u_t^{i*}(v_{t-1}^i, \epsilon_{t-1})$ . La politique de turbinage approximative associée à l'état  $(v_{t-1}, \epsilon_{t-1})$  peut être obtenue par interpolation sur les sommets du simplexe :

$$\hat{u}_t(v_{t-1}, \epsilon_{t-1}) = \sum_{i=1}^{n+1} \lambda_i u_t^{i*}(v_{t-1}^i, \epsilon_{t-1})$$

où  $\lambda = \begin{pmatrix} X \\ e^T \end{pmatrix}^{-1} \begin{pmatrix} v_{t-1} \\ 1 \end{pmatrix}$ . Cette politique interpolée est réalisable puisque (4.13-4.23) est un domaine convexe.

## 4.6 Expériences numériques

Nous avons analysé le compromis entre le temps de calcul et la précision de la méthode de PDS approximative proposée dans ce chapitre à l'aide de 45 problèmes test simulés. Nous avons choisi un horizon d'une année avec pas de temps hebdomadaire. Pour chaque semaine, l'algorithme CFK a été utilisé pour décomposer le pavé initial en simplexes, puis l'algorithme de division de simplexes présenté dans le chapitre précédent a été exécuté pour construire la grille sur laquelle la fonction de valeur est évaluée en résolvant, pour chaque point, le programme linéaire (4.12-4.23).

Les simplexes initiaux ont été générés par l'algorithme CFK pour deux raisons. Dans cette configuration à cinq réservoirs (figure 4.3), la méthode de rehaussement décomposerait le pavé initial en 3840 simplexes initiaux qui devraient être traités à chaque itération de l'algorithme de raffinement de la grille des stocks, alors qu'avec l'algorithme

CFK, ce nombre est de seulement 120 simplexes initiaux (pour un rappel, voir la section 3.3.3). Étant donné que notre objectif est d'illustrer le compromis entre le temps de calcul et la précision obtenue en raffinant la grille, nous avons préféré d'avoir un nombre moins élevé de simplexes initiaux. De plus, avec la méthode de rehaussement, la grille initiale est composée de 243 points ( $3^5$ ). Comme le montreront les résultats, avec tous ces points initiaux, il serait impossible d'illustrer le compromis avec tous les seuils d'imprécision que nous avons utilisés.

Cinquante-deux (52) fonctions de valeur de l'eau ont été construites. La valeur terminale a été supposée nulle. Plusieurs seuils sur l'erreur relative,  $\eta = \frac{\Delta_k}{\Delta_1}$ , où  $k$  est un numéro d'itération, ont été utilisés comme critère d'arrêt. Pour illustrer le compromis entre les efforts de calcul et la précision, les seuils de tolérance variaient entre 0.8 et 0.01.

Le cadre expérimental de ces expériences est discuté dans la section 4.6.1. Les résultats et les analyses sont exposés dans la section 4.6.2.

### 4.6.1 Ensemble test

Nous avons implanté un modèle sans variable hydrologique. Le programme dynamique simplifié (4.7-4.11) s'écrit alors :

$$V_{t-1}(v_{t-1}) = \text{Max}_{u_t, x_t} \left\{ \sum_{i \in I_{RC}} \bar{f}_{it}(u_{it}, v_{i,t-1}) + \sum_{i \in I_F} f_{it}(u_{it}) + E_{Q_t} [V_t(v_t^*(x_t + Q_{I_R t}))] \right\}$$

$$\text{S.l.c. } x_t = v_{t-1} - B_{I_R} u_t$$

$$B_{I_F} u_t = 0$$

$$x_t \geq \underline{v}_t$$

$$\underline{u}_t \leq u_t \leq \bar{u}_t$$

Les paramètres du modèle ont été générés par simulation de Monte Carlo, dont les valeurs, indiquées dans la table 4.1, ont été tirées de distributions uniformes et indépendantes de support de la forme  $[a, b]$ .

La table 4.2 montre le plan de simulation global. Les problèmes test ont été résolus pour la configuration à cinq réservoirs illustrée dans la figure 4.3.

Nous avons supposé que les apports suivaient une distribution log-normale dont les paramètres ont été tirés aléatoirement de la manière suivante :



TABLE 4.1 – Bornes numériques sur les paramètres du modèle à 5 réservoirs

Paramètre	$a$	$b$
$\underline{v}_{it}, i \in I_R$	150	600
$\bar{v}_{it}, i \in I_R$	800	7000
$\underline{u}_{it}, i \in I_R$	0	0
$\bar{u}_{it}, i \in I_R$	$0.05\bar{v}_{it}$	$1.5\bar{v}_{it}$
$\bar{u}_{it}, i \in I_F$	950	3500

TABLE 4.2 – Nombre de duplications par valeur de  $\eta$

$\eta$	# de duplications
0.8	10
0.6	10
0.4	10
0.2	9
0.1	5
0.01	1
# total de duplications	45

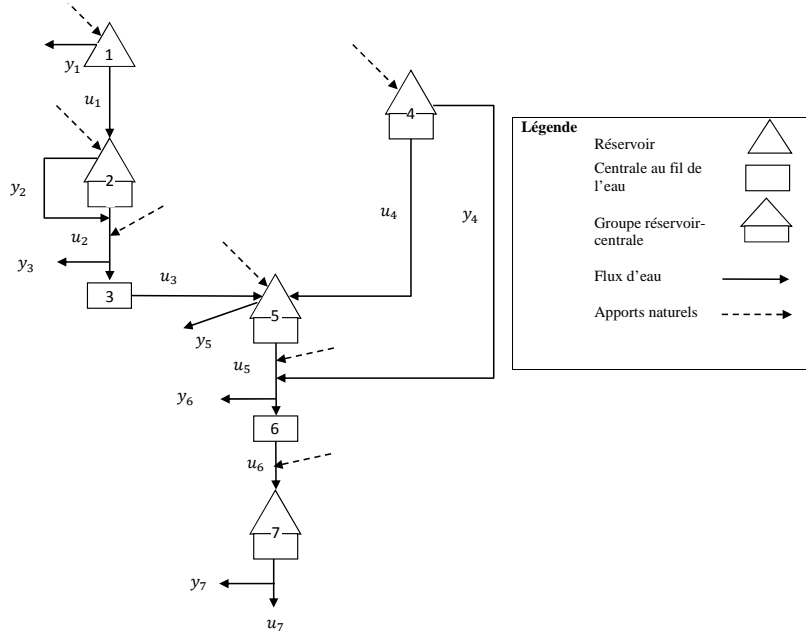


FIGURE 4.3 – Configuration à 5 réservoirs utilisé dans les simulations

$$(\mu_t, \sigma_t) = \begin{cases} (\psi_t, 0.01\psi_t) & \text{avec probabilité } 1/4, \text{ où } \psi_t = 1.2 \ln \max\{\bar{v}_{it}\}, i \in I_R \\ (\varrho_t, 0.04\varrho_t) & \text{avec probabilité } 1/4, \text{ où } \varrho_t = 1.1 \ln \max\{\bar{v}_{it}\}, i \in I_R \\ (\iota_t, 0.015\iota_t) & \text{avec probabilité } 1/4, \text{ où } \iota_t = 0.6 \ln \max\{\bar{v}_{it}\}, i \in I_R \\ (\zeta_t, 0.02\zeta_t) & \text{avec probabilité } 1/4, \text{ où } \zeta_t = 0.4 \ln \max\{\bar{v}_{it}\}, i \in I_R \end{cases}$$

Nous avons choisi des coefficients de bassin (répartition relative des apports) stationnaires  $q = [0.20 \ 0.10 \ 0.30 \ 0.25 \ 0.15]$ .

Finalement, les fonctions de production ont été représentées par des fonctions théoriques de la forme

$$f_i(u_{it}) = \beta_i[(u_{it} + \gamma_i)^{\alpha_i} - \gamma_i^{\alpha_i}], i \in I_F, 0 \leq \alpha_i \leq 1, \beta_i > 0, \gamma_i \geq 0$$

et

$$f_i(u_{it}, v_{it}) = \beta_i[(u_{it} + \gamma_i)^{\alpha_i} - \gamma_i^{\alpha_i} + (u_{it}v_{it})^{\theta_i}], i \in I_{RC}, 0 \leq \alpha_i \leq 1, \beta_i > 0, \gamma_i \geq 0, 0 \leq \theta_i \leq 1$$

La figure 4.4 présente un exemple de chacune de ces fonctions.

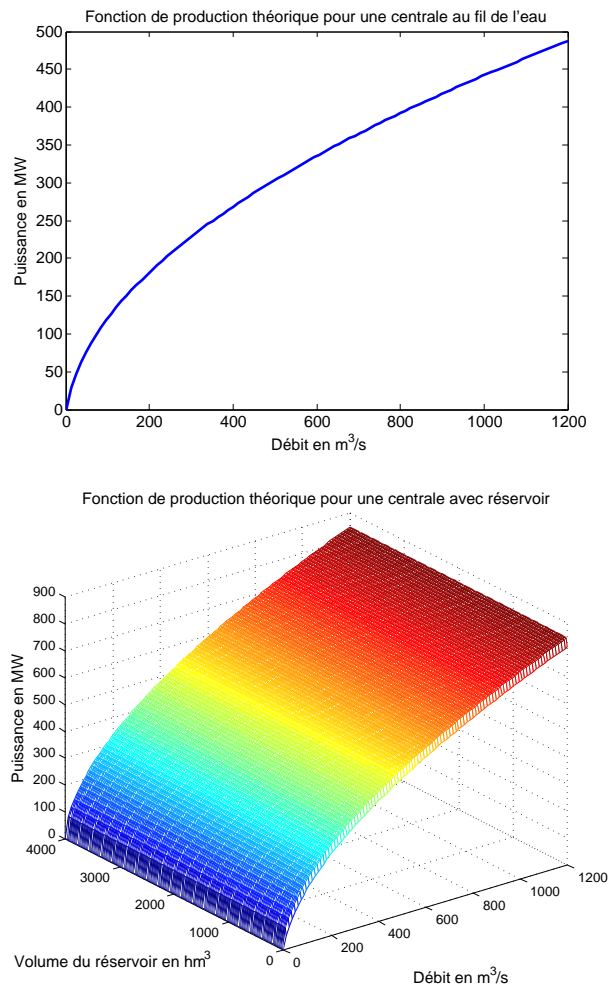


FIGURE 4.4 – Deux exemples de fonctions de production utilisées dans les simulations

## 4.6.2 Résultats et analyses

Les algorithmes ont été implantés avec la version *R2012a* de Matlab et les problèmes linéaires résolus avec Cplex 12.5.1.0. Pour chaque simulation exécutée, nous avons mesuré le temps total de calcul pour construire les 52 fonction de valeur de l'eau. Nous avons ensuite calculé, pour chaque valeur de l'erreur relative ( $\eta$ ), le temps d'exécution moyen ainsi que les valeurs minimales et maximales. Ces statistiques sont indiquées dans la table 4.3.

Ces résultats illustrent bien le compromis entre le temps de calcul et la précision. En effet, avec une tolérance d'erreur initiale  $\eta = 0.8$ , il a fallu, en moyenne, 120.205 s pour approximer la fonction de valeur sur les 52 semaines. Ce temps variait entre 51.210 et 257.759 s.

Pour réduire la tolérance d'erreur initiale de 10% ( $\eta = 0.7$ ), le temps moyen a été multiplié par à peu près 1.438, soit des efforts de calcul de l'ordre de 173 s. Avec une réduction de 40% de l'imprécision initiale, le temps moyen d'exécution a été multiplié par environ 7, soit près de 842 s. Enfin, une réduction de 70% de l'erreur a entraîné, en moyenne, des efforts de calcul 42 fois plus élevés (environ 5 000 s).

TABLE 4.3 – Statistiques sur le temps d'exécution (en secondes) des simulations

$\eta$	Moyenne	Minimum	Maximum
0.8	120.205	51.210	257.759
0.7	172.911	74.166	362.775
0.6	440.639	70.088	1 161.306
0.4	842.463	137.58	1 711.899
0.2	1 498.090	982.471	2 073.618
0.1	4 995.312	3 292.068	8 089.491
0.01	14 622.483	14 622.483	14 622.483

La figure 4.5 montre le temps d'exécution moyen en fonction de la précision ( $1 - \eta$ ) dans un repère semi logarithmique. On observe que la relation est linéaire par morceaux, ce qui suggérerait que le temps de calcul moyen augmente de manière exponentielle avec la précision, mais à des taux de « croissance locale » différents.

Le compromis entre les efforts de calcul et la précision peut être analysé aussi par le nombre moyen de points de grille à chaque période (table 4.4).

Pour une erreur relative initiale  $\eta = 0.8$ , sur l'ensemble des exécutions des simulations, le nombre moyen de points de grille (par période) variait entre 55 et 176.

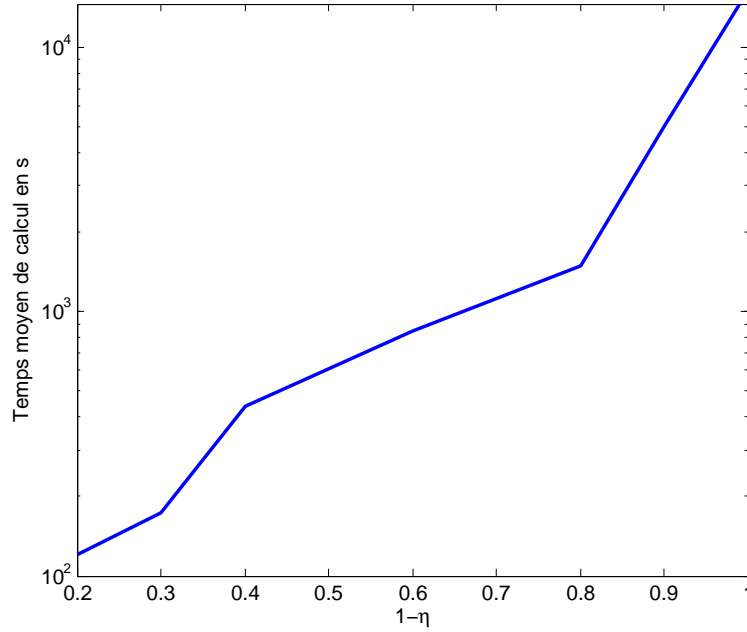


FIGURE 4.5 – Temps moyen de résolution des problèmes en fonction de l'imprécision dans un repère semi-log

En réduisant l'erreur relative initiale de 10%, en moyenne, à chaque période, les grilles comprenaient entre 72 et 297 points. Avec une réduction de 40%, le nombre de points de grille minimal a plus que doublé et le nombre maximal a été multiplié par environ quatre. Pour une erreur relative de 10%, le nombre moyen de points de grille minimal et maximal par période variait entre 301 et 1152, soit une multiplication par environ cinq et sept respectivement.

TABLE 4.4 – Statistiques sur le nombre moyen de points de grille par période

$\eta$	Moyenne	Minimum	Maximum
0.8	103.229	55.481	176.115
0.7	156.431	72.346	297.019
0.6	278.871	95.615	430.808
0.4	409.002	123.115	714.673
0.2	412.546	266.096	568.346
0.1	656.035	301.192	1151.750
0.01	675.115	675.115	675.115

## 4.7 Conclusion

Nous avons discuté dans ce chapitre d'une méthode de PDS approximative basée sur la méthodologie de la partition de l'espace des stocks d'eau en simplexes développé dans le chapitre précédent. Nous sommes revenus sur la modélisation du problème des réservoirs en insistant sur le processus information-décision.

Exploitant l'hypothèse de corrélation spatiale parfaite entre les apports naturels, souvent utilisée dans les cas pratiques par les hydrologues, nous avons développé des formes analytiques pour l'espérance de la fonction de valeur, ce qui contribue à réduire les efforts de calcul. Ensuite, nous avons discuté, de façon détaillée, d'une approche d'optimisation par la programmation linéaire généralisée qui a été partiellement abordée dans le chapitre précédent.

Des expériences numériques réalisées à partir de 45 problèmes test simulés ont permis d'illustrer le compromis entre la complexité des efforts de calcul et la précision. Pour chaque problème, nous avons mesuré le temps de calcul total pour construire la fonction de valeur, sur un horizon de 52 semaines, et pour plusieurs niveaux d'imprécision. Nous avons aussi présenté des statistiques sur le nombre de points de grille moyen par période pour chaque valeur de l'erreur relative.

La méthodologie d'approximation simpliciale sera testée dans un contexte industriel au prochain chapitre.



# Chapitre 5

## Étude de cas : Rio Tinto Alcan

### 5.1 Introduction

Ce chapitre traite d'une application de la méthode de PDS approximative exposée dans les deux chapitres précédents dans le contexte de Rio Tinto Alcan (RTA), une entreprise multinationale de production d'aluminium qui possède des centrales de production d'électricité situées au Saguenay.

Le système hydroélectrique est composé de six centrales de production, dont trois au fil de l'eau, et des barrages construits sur les rivières Péribonka et du Saguenay (figure 5.1). La capacité installée, i.e. si tous les groupes turbo-alternateurs tournent à plein régime, est de 3100 *MW*. Cependant, la production moyenne est d'environ 2100 *MW*.

Ce système alimente quatre alumineries pour une charge totale d'à peu près 2300 *MW*. Pour pallier le manque d'énergie, RTA a plusieurs contrats d'approvisionnement avec Hydro-Québec, dont un de 3 *TWh* allant du 1<sup>er</sup> décembre au 30 novembre.

Depuis les années 70, des outils déterministes sont utilisés pour la planification des opérations du système. Pour chaque journée, un ensemble de scénarios d'apports est construit, et un problème d'optimisation est résolu pour chaque scénario. La médiane des solutions est utilisée pour la journée. Pour la journée suivante, les informations sont mises à jour (par ex. à partir des données météo), puis de nouveaux scénarios sont construits, et le modèle est réoptimisé à chaque fois.

RTA s'intéresse à l'application de modèles stochastiques à son système, notamment pour les raisons évoquées dans le chapitre 1 sur les désavantages des modèles déterministes. Cette application s'inscrit dans cette perspective.

La section 5.2 est une description du système à l'étude. Dans la section 5.3, l'approxima-

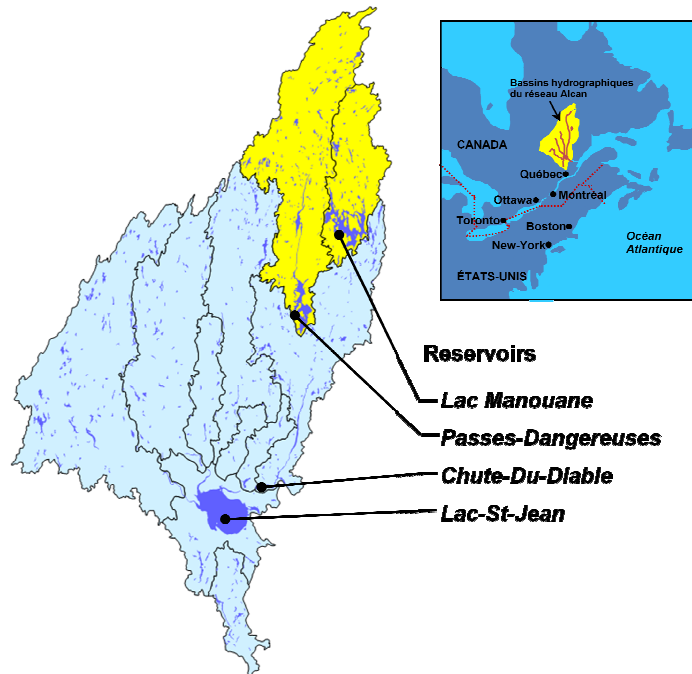


FIGURE 5.1 – Bassin hydrographique du système de RTA

tion par la programmation linéaire généralisée discutée dans la section 4.5 est adaptée au contexte de RTA. Ensuite, le calcul de l’espérance de la fonction de valeur est établi dans la section 5.4, en se basant sur le processus information-décision du modèle de RTA. Finalement, un cadre expérimental d’étude est décrit dans la section 5.5. Des résultats et une comparaison avec un modèle mis au point par RTA y sont aussi analysés.

## 5.2 Description du système à l’étude

Pour des raisons opérationnelles, RTA s’intéresse à des études sur la partie sud de son système illustrée dans la figure 5.2. Les flèches pleines représentent les débits turbinés, celles en traitillés les débits délestés et celles en pointillés les apports naturels.

Le système aval est sous-dimensionné par rapport aux apports qu’il reçoit et la gestion est beaucoup plus difficile que celle du système amont. De plus, la capacité de turbinage du système amont est assez faible. Le système aval comprend deux réservoirs : Chute-Du-Diable et Lac Saint-Jean associés respectivement aux centrales Chute-Du-Diable et Isle-Maligne, et deux centrales au fil de l’eau : Chute-Savanne et Shipshaw.

Pour tout nœud  $i, i = 1, \dots, 4$ , du système, comme dans les deux chapitres précédents, notons respectivement par  $u_{it}$  et  $v_{it}$  le volume d’eau turbiné et le volume initial du



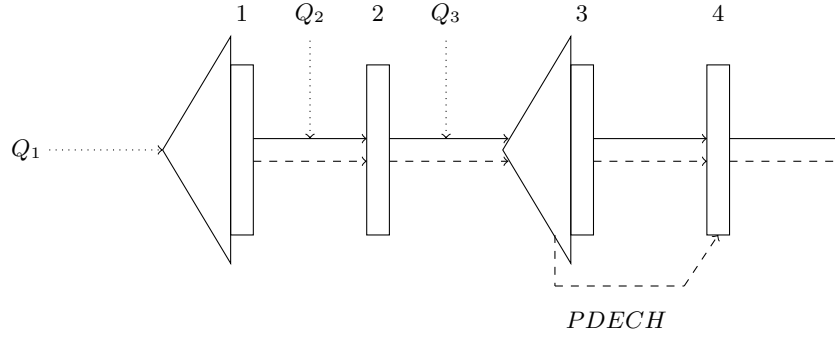


FIGURE 5.2 – Représentation du sous-système étudié

réservoir associé à la période  $t$ , et par  $f_i(v_{it}, v_{i,t+1}, u_{it})$  ou  $f_i(u_{it})$ , la production de la centrale  $i$  à la même période. Les apports naturels qui arrivent dans le système à la période  $t$  sont notés par  $Q_t$ .

Dans ce qui suit, les données utilisées sont normalisées pour des raisons de confidentialité.

Pour les centrales Chute-Du-Diable (CCD, nœud 1) et Isle-Maligne (CIM, nœud 3) (associées aux réservoirs portant les mêmes numéros), la puissance dépend du débit turbiné et du niveau des réservoirs. Bien que la centrale Chute-Savanne (CCS, nœud 2) soit au fil de l'eau, en plus du débit turbiné, sa puissance dépend du niveau du réservoir Lac Saint-Jean (LSJ, nœud 3), car celui-ci influence le niveau de son bief aval. La figures 5.3 illustre la puissance des trois premières centrales en fonction du débit pour différents niveaux de stocks. Ces fonctions sont non linéaires et non concaves.

Enfin, la puissance de la centrale Shipshaw (CSH) (nœud 4), illustrée dans la figure 5.4, dépend uniquement du débit turbiné.

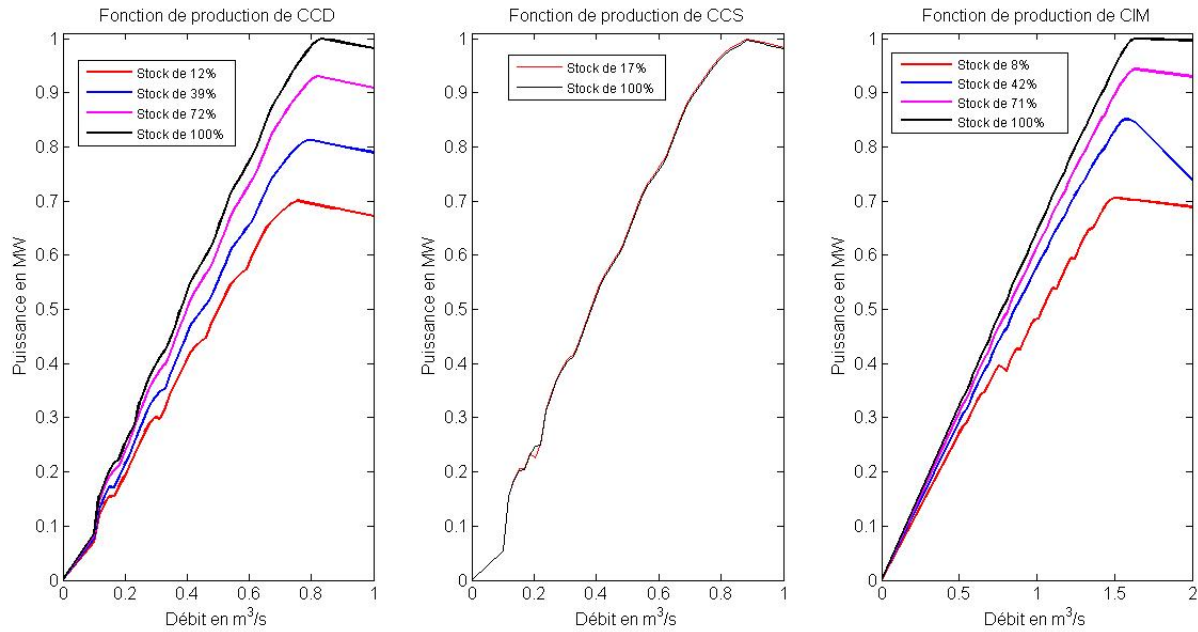


FIGURE 5.3 – Fonctions de production des centrales CCD, CCS et CIM en fonction des débits

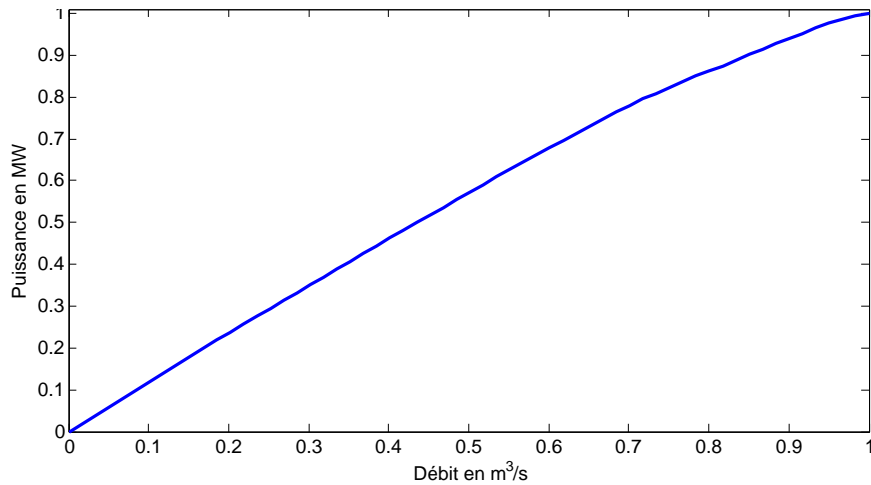


FIGURE 5.4 – Fonction de production de la centrale SH

Le débit maximal turbinable dans les centrales Chute-Du-Diable et Isle-Maligne est une fonction du volume d'eau stocké en début de période, tandis qu'il est constant pour les centrales Chute-Savanne et Shipshaw.  $\tilde{B}_i(v_{it}), i = 1, 3$ , dénotent ces fonctions de turbinage maximal. La figure 5.5 montre les débits turbinables maximaux en fonction du stock de début de période pour les deux premières centrales.

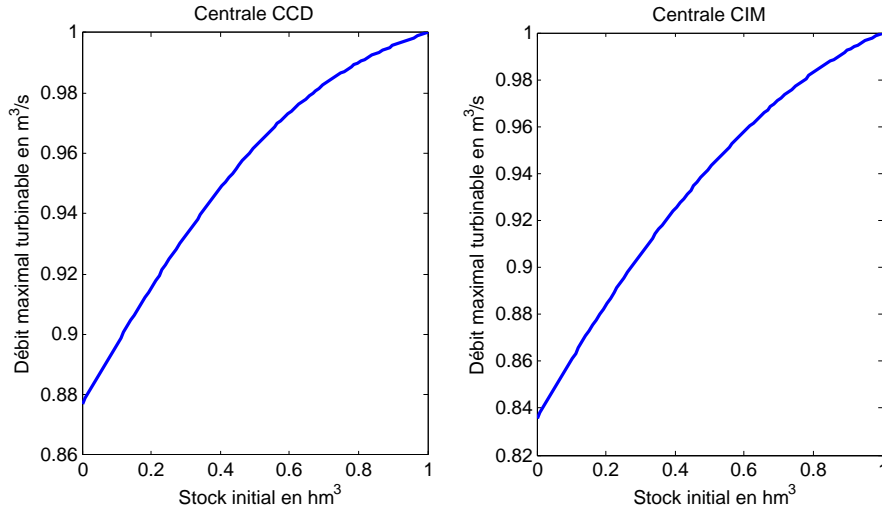


FIGURE 5.5 – Débit maximal en  $m^3/s$  des centrales CD et IM en fonction du stock initial

Dans les quatre centrales, tout surplus d'eau est délesté et produit une perte de puissance, car le délestage augmente le niveau du bief aval. Nous notons par  $y_{it}$  le volume d'eau délesté à la centrale  $i$  à la période  $t$ . Le réservoir du lac Saint-Jean est muni d'une autre issue de délestage (PDECH). L'eau qui y passe n'influence pas la production. Toutefois, cet évacuateur a un débit maximal qui dépend du volume d'eau stocké (figure 5.6).

À la période  $t$ ,  $y_{3,1t}$  dénote les flux d'eau délestés par la centrale IM, et  $y_{3,2t}$ , les flux délestés par l'évacuateur PDECH.  $\tilde{D}(v_{3t})$  dénote le débit maximal pouvant être délesté par l'issue PDECH. La figure 5.7 montre la perte de puissance pour les quatre centrales en fonction du débit délesté. On voit que ces pertes peuvent facilement être approximées par des fonctions linéaires.  $\tilde{A}_i(y_{it}), i = 1, \dots, 4$ , dénotent ces approximations.

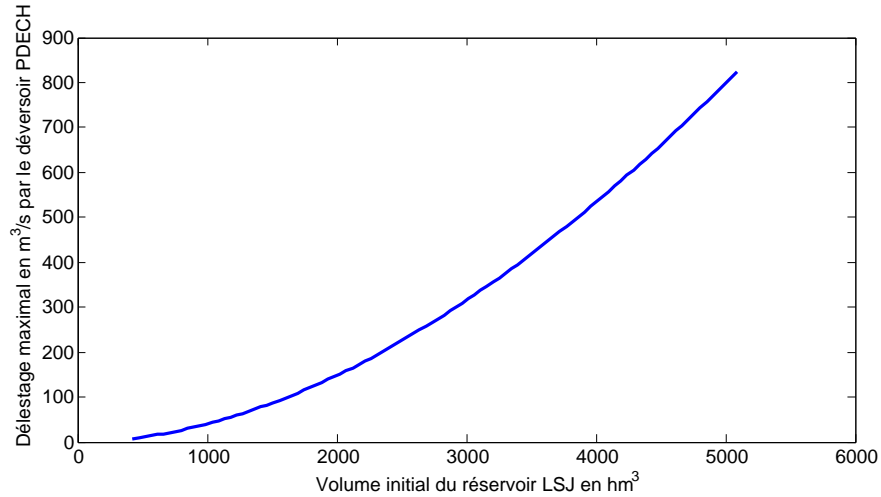


FIGURE 5.6 – Délestage maximal en  $m^3/s$  par l'issue PDECH en fonction du volume d'eau stocké au LSJ

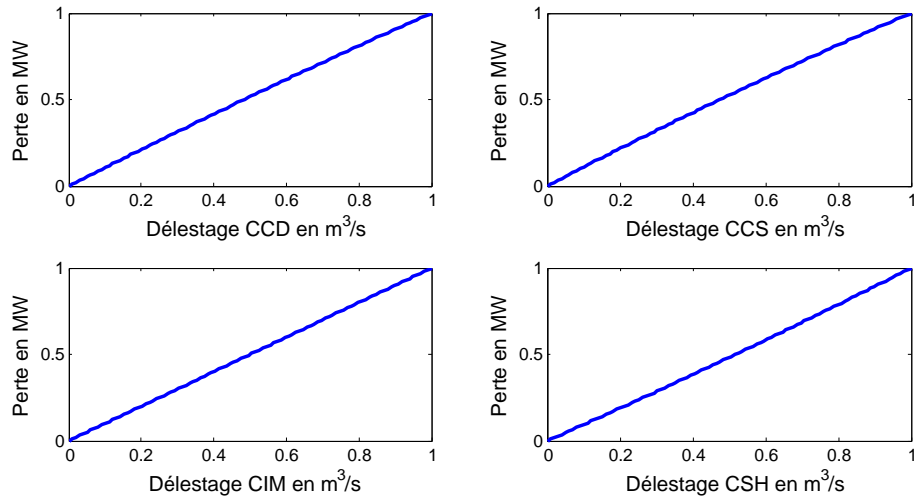


FIGURE 5.7 – Fonction de perte de puissance en  $MW$

La politique de gestion du système peut être obtenue par un modèle de programmation dynamique stochastique (PDS). L'horizon de planification est l'année et le pas de temps la semaine. Les apports naturels à la semaine  $t - 1$ ,  $Q_{t-1}$ , sont la variable hydrologique à la semaine  $t$ . Nous donnons des détails sur ce choix à la section 5.4.

Pour  $t = 52, 51, \dots, 1$ , l'équation de récurrence est  $V_t(v_t, Q_{t-1}) = E_{Q_t|Q_{t-1}}[G_t(v_t, Q_t)]$ , avec

$$G_t(v_t, Q_t) = \max_{u_t, y_t} \left\{ \sum_{i=1,3} f_i(v_{it}, v_{i,t+1}, u_{it}) + f_2(v_{3t}, v_{3,t+1}, u_{2t}) + f_4(u_{4t}) - \sum_{j=1,2,4} \tilde{A}_j(y_j) - \tilde{A}_3(y_{3,1t}) + V_{t+1}(v_{t+1}, Q_t) \right\}$$

S.l.c.  $v_{t+1} = v_t + Q_{IRt} - B_{IR}u_t - C_{IR}y_{It}$

$$\underline{v}_{t+1} \leq v_{t+1} \leq \bar{v}_{t+1}$$

$$0 \leq u_{it} \leq \tilde{B}_i(v_{it}), \quad i = 1, 3$$

$$u_{1t} + y_{1t} + Q_{2t} - u_{2t} - y_{2t} = 0$$

$$u_{3t} + y_{3,1t} + y_{3,2t} - u_{4t} - y_{4t} = 0$$

$$0 \leq u_{it} \leq \bar{u}_i, \quad i = 2, 4$$

$$y_{3,2t} \leq \tilde{D}(v_{3t})$$

$$y_t \geq 0$$

Si à la période  $t$  les stocks et les apports naturels sont faibles, il peut être impossible de satisfaire les bornes à la période suivante comme le montre la figure 5.8 pour le lac Saint-Jean. En effet, au cours de l'été le niveau du lac doit être assez élevé pour permettre la navigation et les activités de récréation. Si les apports naturels sont faibles à la fin du printemps, le problème peut donc être irréalisable.

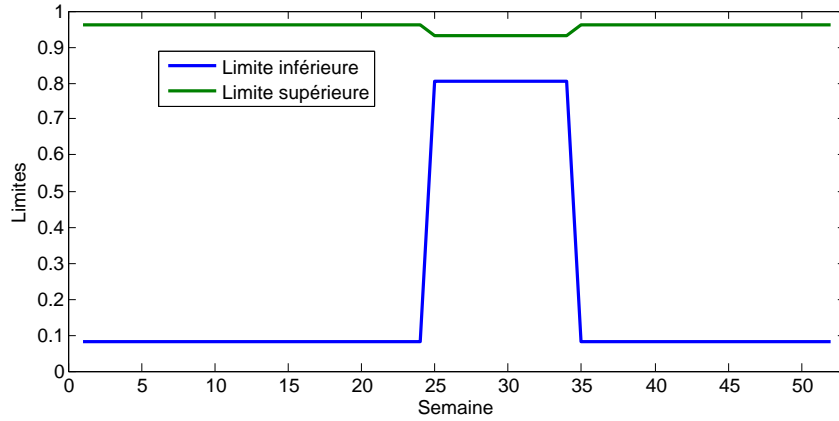


FIGURE 5.8 – Limites du réservoir LSJ

Pour éviter des cas de non réalisabilité, la violation des contraintes de bornes est pénalisée dans l'objectif. Pour chaque réservoir, chaque  $hm^3$  d'eau en-dessous ou en-dessus du

niveau réglementaire est pénalisé de 0.18 (coefficient de pénalité utilisé par RTA). Les pénalités sont donc de la forme  $-0.18\max\{0, \underline{v}_{t+1} - v_{t+1}\}$  et  $-0.18\max\{0, v_{t+1} - \bar{v}_{t+1}\}$ .

Le problème pénalisé s'écrit alors :

$$G_t(v_t, Q_t) = \max_{u_t, y_t} \left\{ \sum_{i=1,3} f_i(v_{it}, v_{i,t+1}, u_{it}) + f_2(v_{3t}, v_{3,t+1}, u_{2t}) + f_4(u_{4t}) - \sum_{j=1,2,4} \tilde{A}_j(y_j) - \tilde{A}_3(y_{3,1t}) - 0.18\tilde{d}_{1t} - 0.18\tilde{d}_{2t} + V_{t+1}(v_{t+1}, Q_t) \right\} \quad (5.1)$$

$$\text{S.l.c. } v_{t+1} = v_t + Q_{IRt} - B_{IR}u_t - C_{IR}y_t \quad (5.2)$$

$$\tilde{d}_{1t} \geq \underline{v}_{t+1} - v_{t+1} \quad (5.3)$$

$$\tilde{d}_{2t} \geq v_{t+1} - \bar{v}_{t+1} \quad (5.4)$$

$$0 \leq u_{it} \leq \tilde{B}_i(v_{it}), i = 1, 3 \quad (5.5)$$

$$u_{1t} + y_{1t} + Q_{2t} - u_{2t} - y_{2t} = 0 \quad (5.6)$$

$$u_{3t} + y_{3,1t} + y_{3,2t} - u_{4t} - y_{4t} = 0 \quad (5.7)$$

$$0 \leq u_{it} \leq \bar{u}_i, i = 2, 4 \quad (5.8)$$

$$y_{3,2t} \leq \tilde{D}(v_{3t}) \quad (5.9)$$

$$y_t \geq 0 \quad (5.10)$$

$$\tilde{d}_{1t}, \tilde{d}_{2t} \geq 0 \quad (5.11)$$

### 5.3 Formulation linéaire du programme dynamique

Le programme (5.1-5.11) est non linéaire et non convexe à cause de la non linéarité et la non concavité des fonctions de production.

Pour chaque centrale  $i, i = 1, \dots, 4$ , indépendamment de la semaine, les fonctions de production ont été approximées sur des échantillons de points  $\{(\hat{u}_{ij}, \hat{v}'_{ij}) \mid j \in \Gamma_i, i = 1, 3\}$  pour les centrales Chute-Du-Diable et Isle-Maligne,  $\{(\hat{u}_{2j}, \hat{v}'_{3j}) \mid j \in \Gamma_2\}$  pour la centrale Chute-Savanne, et  $\{\hat{u}_{4j} \mid j \in \Gamma_4\}$  pour la centrale Shipshaw. Rappelons que la puissance de la centrale Chute-Savanne est aussi fonction du niveau du LSJ. La stratégie de discrétisation utilisée est exposée en détail dans la section 5.3.1.

À la semaine  $t$ , nous disposons aussi des évaluations d'une approximation  $\hat{V}_{t+1}(\hat{v}_{t+1}, Q_t)$  de la fonction de valeur  $V_{t+1}(v_{t+1}, Q_t)$  sur une grille de points  $\{\hat{v}_{k,t+1} \mid k \in \Upsilon_t\}$ . Cette grille est construite en partitionnant l'espace des stocks réalisables en simplexes par la

méthode présentée dans la section 3.3.4.

Pour permettre les pénalisations, nous rajoutons deux points à la grille : le stock nul et un stock plus grand que la borne supérieure  $\bar{v}_{t+1}$ . Comme pour tous les autres points de la grille, la fonction de valeur est évaluée au stock nul. Pour tout stock  $\hat{v}_{t+1} > \bar{v}_{t+1}$ , nous supposons que  $\hat{V}_{t+1}(\hat{v}_{t+1}, Q_t) = \hat{V}_{t+1}(\bar{v}_{t+1}, Q_t)$ .

Pour tenir compte des variations de stocks au cours de la semaine dans l'évaluation de la production, nous utiliserons la moyenne des stocks de début et de fin de semaine. Ces stocks moyens sont notés par  $v'_t$ , i.e.  $v'_t = 0.5(v_t + v_{t+1})$ .

Enfin, rappelons que les fonctions de perte sont remplacées par des approximations linéaires  $\tilde{A}_j(y_j), j = 1, \dots, 4$ . Pour un niveau initial de stock donné  $v_t$ , les débits maximaux turbinables  $\tilde{B}_i(v_{it}), i = 1, 3$  ainsi que le débit maximal pouvant être délesté par l'issue PDECH  $\tilde{D}(v_{3t})$  sont des constantes.

Par la programmation linéaire généralisée (voir section 4.5), pour  $t = 52, 51, \dots, 1$ , la récurrence suivante est une approximation linéaire du programme (5.1-5.11) :

$$\begin{aligned} \hat{G}_t(v_t, Q_t) = \max_{u_t, y_t, \lambda, \mu} \left\{ \sum_{i=1,3} \sum_{j \in \Gamma_i} f_i(\hat{v}'_{ij}, \hat{u}_{ij}) \lambda_{ij} + \sum_{j \in \Gamma_2} f_2(\hat{v}'_{3j}, \hat{u}_{2j}) \lambda_{2j} + \sum_{j \in \Gamma_4} f_4(\hat{u}_{4j}) \lambda_{4j} - \right. \\ \left. \sum_{s=1,2,4} \tilde{A}_s(y_s) - \tilde{A}_3(y_{3,1t}) - 0.18\tilde{d}_{1t} - 0.18\tilde{d}_{2t} + \right. \\ \left. \sum_{k \in \Upsilon_t} \hat{V}_{t+1}(\hat{v}_{k,t+1}, Q_t) \mu_k \right. \end{aligned} \quad (5.12)$$

$$\text{S.l.c. } v_{t+1} = v_t + Q_{IRt} - B_{IR}u_t - C_{IR}y_t \quad (5.13)$$

$$v_{t+1} = \sum_{k \in \Upsilon_t} \hat{v}_{k,t+1} \mu_k \quad (5.14)$$

$$\sum_{k \in \Upsilon_t} \mu_k = 1 \quad (5.15)$$

$$v'_t - 0.5v_{t+1} = 0.5v_t \quad (5.16)$$

$$v'_{it} = \sum_{j \in \Gamma_i} \hat{v}'_{ij} \lambda_{ij}, \quad i = 1, 3 \quad (5.17)$$

$$v'_{2t} = \sum_{j \in \Gamma_2} \hat{v}'_{3j} \lambda_{2j} \quad (5.18)$$

$$\tilde{d}_{1t} \geq v_{t+1} - v_{t+1} \quad (5.19)$$

$$\tilde{d}_{2t} \geq v_{t+1} - \bar{v}_{t+1} \quad (5.20)$$

$$0 \leq u_{it} \leq \tilde{B}_i(v_{it}), \quad i = 1, 3 \quad (5.21)$$

$$u_{1t} + y_{1t} + Q_{2t} - u_{2t} - y_{2t} = 0 \quad (5.22)$$

$$u_{3t} + y_{3,1t} + y_{3,2t} - u_{4t} - y_{4t} = 0 \quad (5.23)$$

$$u_{it} = \sum_{j \in \Gamma_i} \hat{u}_{ij} \lambda_{ij}, \quad i = 1, \dots, 4 \quad (5.24)$$

$$\sum_{j \in \Gamma_i} \lambda_{ij} = 1, \quad i = 1, \dots, 4 \quad (5.25)$$

$$0 \leq u_{it} \leq \bar{u}_i, \quad i = 2, 4 \quad (5.26)$$

$$y_{3,2t} \leq \tilde{D}(v_{3t}) \quad (5.27)$$

$$y_t \geq 0 \quad (5.28)$$

$$\tilde{d}_{1t}, \tilde{d}_{2t} \geq 0 \quad (5.29)$$

$$\lambda, \mu \geq 0 \quad (5.30)$$

### 5.3.1 Discrétisation des débits turbinés

Ici, notre objectif est de construire les grilles  $\{(\hat{u}_{ij}, \hat{v}'_{ij}) \mid j \in \Gamma_i, i = 1, 3\}$  pour les centrales Chute-Du-Diable et Isle-Maligne,  $\{(\hat{u}_{2j}, \hat{v}'_{3j}) \mid j \in \Gamma_2\}$  pour la centrale Chute-Savanne, et  $\{\hat{u}_{4j} \mid j \in \Gamma_4\}$  pour la centrale Shipshaw mentionnées dans la section



précédente. Les stocks mentionnés dans cette section sont donc des stocks moyens auxquels sont évaluées les fonctions de production qu'il ne faut pas confondre avec les stocks finaux auxquels sont évaluées les fonctions de valeur.

Nous avons reçu de RTA des échantillons de 54 points de débits pour 20 niveaux de stocks différents pour les trois premières centrales. Pour chacune de ces centrales, nous avons donc un échantillon de  $54 \times 20$  points de débits  $\times$  stocks. Nous avons aussi l'évaluation de la puissance en chaque paire débit-stock. Pour la dernière centrale, nous avons reçu les évaluations de la puissance sur un échantillon de 51 points de débits.

Pour chaque centrale et chaque niveau de stock, nous avons, dans un premier temps, choisi comme points de discrétisation « candidats », des débits où l'enveloppe concave coïncide avec la vraie production comme le montrent les figures 5.9 à 5.12.

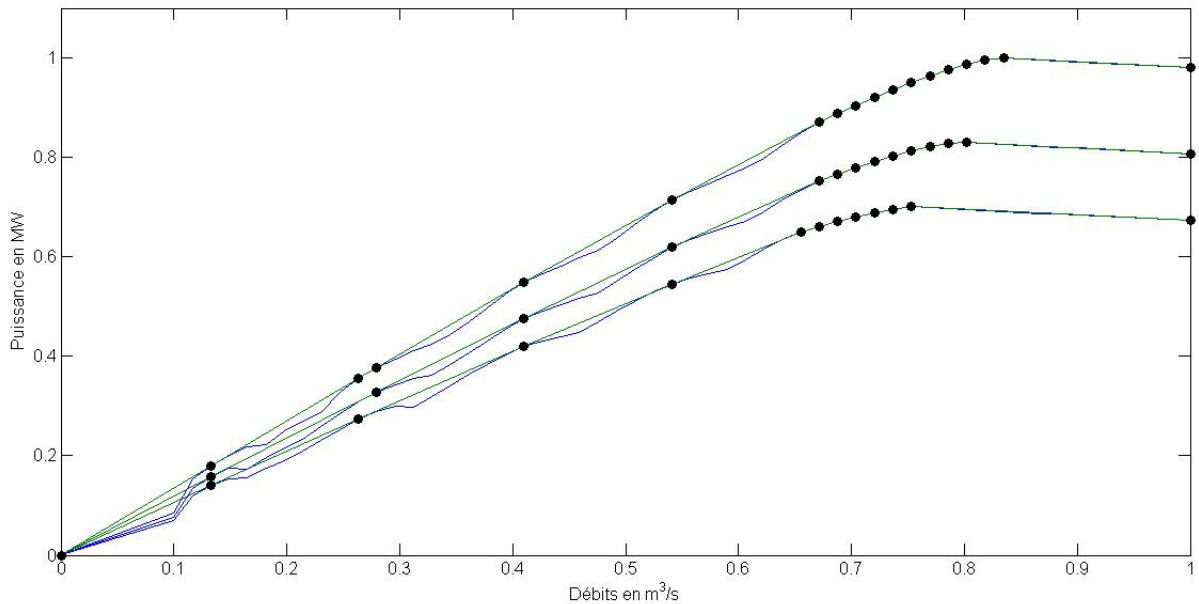


FIGURE 5.9 – Illustration des points de discrétisation candidats des fonctions de production de la centrale CD

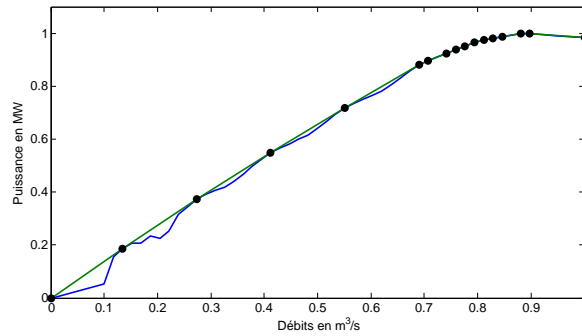


FIGURE 5.10 – Illustration des points de discrétisation candidats des fonctions de production de la centrale CS

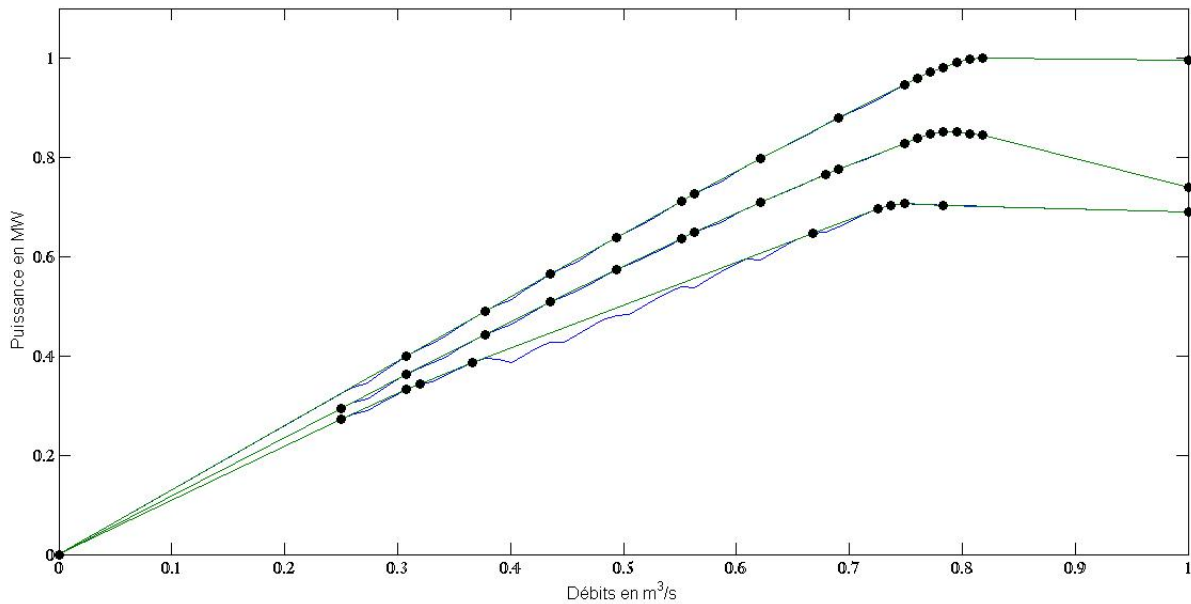


FIGURE 5.11 – Illustration des points de discrétisation candidats des fonctions de production de la centrale IM

Plus le nombre de points de discrétisation des débits est élevé, plus la taille (le nombre de variables) du programme d'optimisation (5.12-5.30) augmente. Ceci peut devenir un enjeu important si le système comprend beaucoup de nœuds, et plus particulièrement si le nombre de réservoirs est élevé. On a donc intérêt à utiliser le moins de points possible. Toutefois, la contrepartie pourrait être une moins bonne approximation des fonctions de production.

Pour construire les grilles d'évaluation des fonctions de production, nous analysons le compromis entre : (i) l'erreur d'interpolation, en  $MW$ , commise en comparant (pour

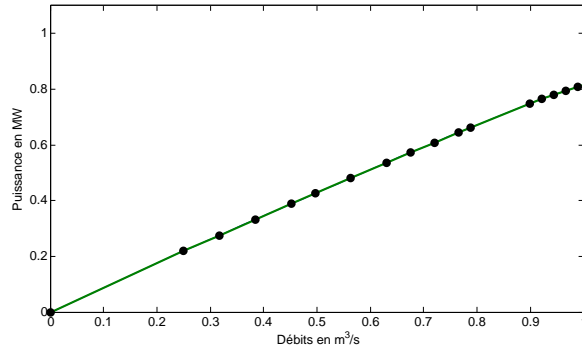


FIGURE 5.12 – Illustration des points de discrétisation candidats des fonctions de production de la centrale SH

chaque niveau de stocks), pour tous les points de débits initiaux, les valeurs interpolées de la production sur la grille de débits retenus et les valeurs exactes ; et (ii) le temps de calcul pour construire les fonctions de valeur de l'eau.

Définissons les stratégies de discrétisation suivantes :

$ST^0$  : pour chaque centrale avec réservoir : pour chaque niveau de stock, parmi les 54 débits initiaux, choisir ceux où l'enveloppe concave coïncide avec la vraie fonction de production. Pour la dernière centrale (au fil de l'eau), parmi les 51 débits initiaux, choisir ceux où l'enveloppe concave coïncide avec la vraie fonction de production.

$ST^1$  : choisir sur les enveloppes concaves les points de changement de pente brusques. Un exemple est montré dans la figure 5.13.

$ST^2$  : tout d'abord, regrouper les stocks « voisins » par paires (en prenant la moyenne). Ensuite, appliquer la stratégie  $ST^1$ .

$ST^3$  : avec la stratégie  $ST^2$ , choisir uniquement les points de changement de pente les plus prononcés. Par exemple, en éliminant le deuxième point dans la figure 5.13.

La stratégie  $ST^0$  a été utilisée uniquement pour des fins de comparaison avec les autres stratégies de réduction de la taille des échantillons de points. Pour chaque stratégie, la table 5.1 indique des statistiques sur l'erreur d'interpolation linéaire commise.  $E_{moy}$  est l'erreur moyenne,  $E_{min}$  et  $E_{max}$  sont respectivement les erreurs minimale et maximale. Le nombre de points de discrétisation (débits  $\times$  stocks) est aussi indiqué.

L'erreur moyenne a été retenue - et non la moyenne des valeurs absolues ni la racine du carré moyen des erreurs -, afin d'observer si, en moyenne, la stratégie de discrétisation sous-estime ou surestime la production.

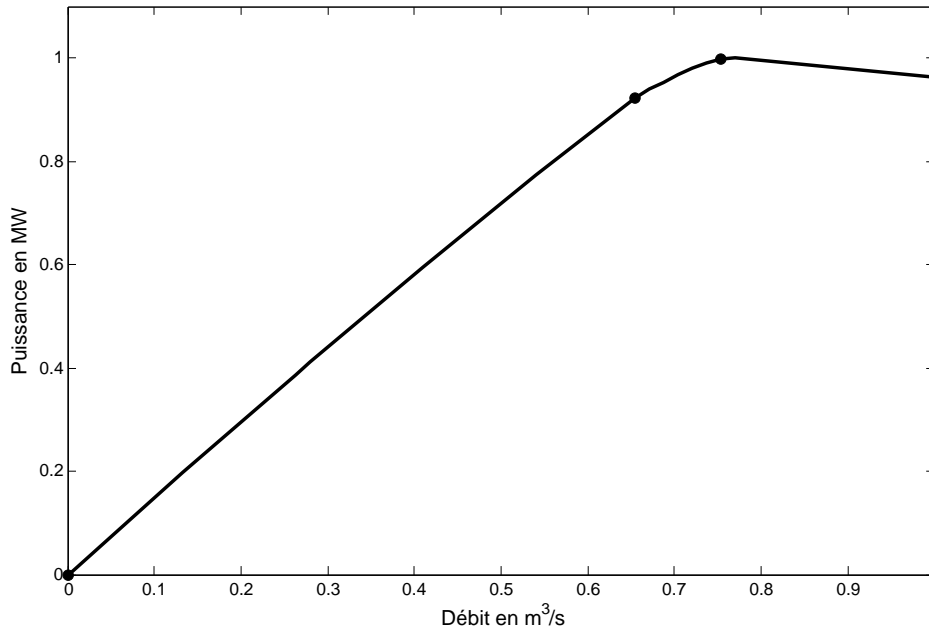


FIGURE 5.13 – Illustration de la discrétisation  $ST^1$

En approximant les fonctions de production par leur enveloppe concave, en moyenne, la production des centrales Chute-Du-Diable et Isle-Maligne est surévaluée de moins de 1.5 MW. Dans le cas de la centrale Chute-Savanne, l'erreur d'interpolation moyenne est d'environ 2.30 MW, et de moins d'un MW pour la centrale Shipshaw. En utilisant l'enveloppe concave, la production est toujours surévaluée.

En réduisant ce nombre de points initial tels que définis par la stratégie  $ST^1$ , dans certains cas, la production est surévaluée et dans d'autres, elle est sous-évaluée. En moyenne, la production de la centrale Shipshaw est sous-évaluée d'à peu près 2.02 MW, et celle des autres centrales de moins d'un MW.

En regroupant les stocks par paire (troisième stratégie), en moyenne, la production de la centrale Chute-Savanne est légèrement surestimée, tandis qu'elle est sous-estimée d'environ 1.53 MW pour la centrale Chute-Du-Diable et d'environ 2.67 MW dans le cas de la centrale Isle-Maligne.

Avec la dernière stratégie, i.e. en réduisant encore le nombre de points obtenus par les regroupements, en moyenne, la production de la centrale Chute-Du-Diable est sous-estimée d'environ 3.15 MW, et celles des trois autres centrales de moins de 5.5 MW. L'effet de réduction de la taille des grilles des débits est plus apparent lorsqu'on compare

les valeurs minimales.

TABLE 5.1 – Statistiques sur l’erreur d’interpolation linéaire pour chaque stratégie de discrétisation

$ST^0$				
Centrale	# total points (débits $\times$ stocks)	$E_{moy}$	$E_{min}$	$E_{max}$
CD	328	1.3933	0.0000	11.3708
CS	340	2.2695	0.0000	20.1139
IM	337	1.1127	0.0000	18.8381
SH	29	0.2783	0.0000	1.6634
Total	1034	–	–	–
$ST^1$				
Centrale	# total points (débits $\times$ stocks)	$E_{moy}$	$E_{min}$	$E_{max}$
CD	97	-0.6090	-6.3065	9.6424
CS	101	-0.3280	-5.7726	19.7847
IM	109	-0.7584	-10.5956	16.9735
SH	7	-2.0177	-8.6249	0.6718
Total	314	–	–	–
$ST^2$				
Centrale	# total points (débits $\times$ stocks)	$E_{moy}$	$E_{min}$	$E_{max}$
CD	47	-1.5353	-13.4415	11.3550
CS	58	0.2239	-9.8307	20.1139
IM	49	-2.6696	-20.1419	14.4038
SH	7	-2.0177	-8.6249	0.6718
Total	161	–	–	–
$ST^3$				
Centrale	# total points (débits $\times$ stocks)	$E_{moy}$	$E_{min}$	$E_{max}$
CD	37	-3.1452	-14.4526	11.2648
CS	42	-5.0267	-13.4606	20.0652
IM	43	-5.2988	-25.9135	14.4038
SH	5	-5.3652	-12.0907	0.0000
Total	127	–	–	–

Pour analyser le deuxième volet de comparaison des différentes stratégies (temps de calcul), pour chaque semaine, nous devons résoudre le programme (5.12-5.30) et ensuite calculer l’espérance de la fonction  $\hat{G}_t$ ; la variable hydrologique  $Q_{t-1}$  n’a pas été incluse dans le modèle (cette question est traitée en détail dans les sections 5.4 et 5.5.) Pour résoudre le programme, nous avons besoin aussi de construire une grilles de stocks finaux

pour chaque semaine comme nous l'avons exposé dans la section précédente. Quatre tolérances d'erreur relative ( $\eta = \frac{\Delta_k}{\Delta_1}$ ) ont été choisies pour estimer les fonctions de valeur (pour un rappel, voir la section 3.3.4) ;  $k$  est un nombre d'itération. Les hyperrectangles initiaux ont été décomposés en simplexes par l'algorithme de rehaussement développé dans la section 3.3.3.

Les temps, mesurés en secondes, sont consignés dans la table 5.2. En général, le temps de résolution est moins important quand moins de points de discrétisation (*débits*  $\times$  *stocks*) sont utilisés. Lorsque les fonctions de production sont approximées par leur enveloppe concave ( $ST^0$  : stratégie avec le plus de points), en moyenne, le temps de résolution est multiplié par plus de 2 comparativement aux autres cas où moins de points ont été utilisés. Ce facteur de multiplication est d'à peu près 2.08 par rapport aux stratégies  $ST^1$  et  $ST^2$  et d'environ 2.31 par rapport à la stratégie où le nombre de points de discrétisation est moins élevée ( $ST^3$ ).

Ces différences seraient vraisemblablement encore plus importantes si on tenait compte de l'autocorrélation des apports (inclusion de la variable hydrologique dans le modèle).

TABLE 5.2 – Temps de résolution en secondes du problème pour chaque stratégie de discrétisation des débits

Stratégie	$\eta = 0.8$	$\eta = 0.5$	$\eta = 0.1$	$\eta = 0.001$	Moyenne
$ST^0$	130.7093	173.8288	637.6737	3 739.6000	1 170.4530
$ST^1$	55.6452	74.1844	283.7806	1 838.7000	563.0776
$ST^2$	50.2090	66.4207	266.5348	1 870.3000	563.3661
$ST^3$	46.4096	60.3900	245.7428	1 670.9000	505.8606

La stratégie  $ST^3$  sera utilisée dans les expériences numériques (voir section 5.5).

## 5.4 Calcul de l'espérance de la fonction $\hat{G}_t$

Les éventuels surplus d'eau sont causés par les apports naturels incertains ( $Q_t$ ) qui arrivent aux trois premiers nœuds du système (figure 5.2). La figure 5.14 montre les apports naturels hebdomadaires totaux historiques sur 33 années, avec des pics à la fin du printemps dûs à la fonte de la neige.

Le processus des apports semble être fortement autocorrélé. En effet, dans l'échantillon mentionné plus haut, l'autocorrélation d'ordre 1 est de 0.895. Ceci motive le choix des apports naturels totaux de la période  $t - 1$  ( $Q_{t-1}$ ) comme variable hydrologique à la période  $t$ .

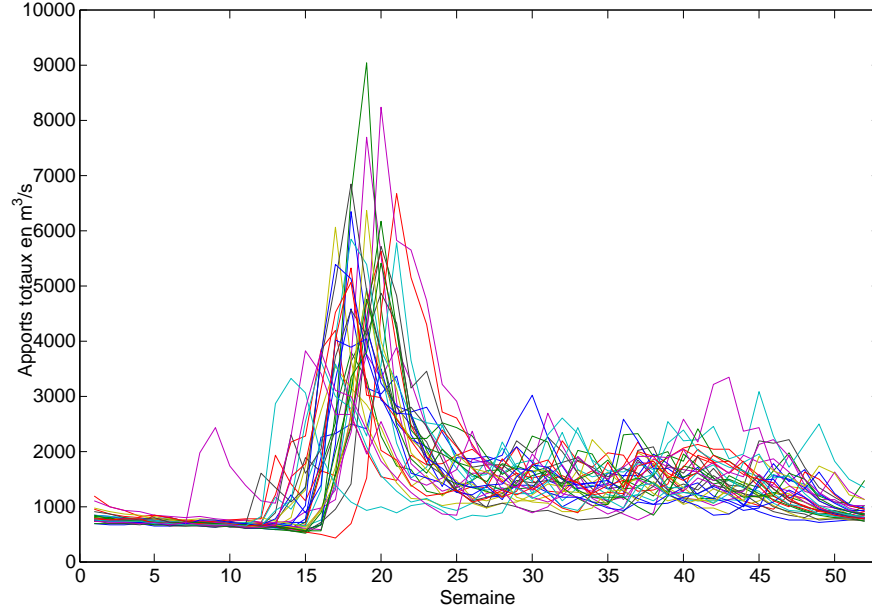


FIGURE 5.14 – Apports naturels hebdomadaires totaux historiques

Contrairement au processus décision-information traité dans le chapitre précédent, le modèle utilisé par RTA suppose que les décisions de turbinage sont prises après avoir observé les apports naturels. Sur recommandation de la compagnie, nous supposons que les apports suivent une distribution log-normale. De plus, on supposera que ceux-ci sont parfaitement corrélés spatialement.

À chaque semaine  $t$ , des coefficients de bassin seront utilisés pour désagréger chaque valeur de  $Q_t$  aux différents sites. Pour un nœud donné, ce coefficient est estimé par le ratio entre les apports historiques moyens de la semaine à ce nœud et les apports moyens historiques totaux de cette même semaine.

On supposera aussi que le processus  $Q_t$  est autorégressif d'ordre 1, ce qui implique qu'il est Markovien. On notera par  $\{q_t^i | 1 \leq i \leq l_t\}$  ses valeurs possibles et  $P(Q_t = q_t^i | Q_{t-1})$  les probabilités de transition. On a donc :

$$\hat{V}_t(v_t, Q_{t-1}) = \sum_{i=1}^{l_t} P(Q_t = q_t^i | Q_{t-1}) \hat{G}_t(v_t, Q_t) \quad (5.31)$$

Les valeurs de  $Q_t$  seront discrétisées et les probabilités de transition estimées à partir des distribution de probabilités correspondantes. Pour chaque valeur de  $Q_{t-1}$ , notons par  $\mu(Q_{t-1})$  et  $\sigma^2(Q_{t-1})$  les paramètres des distributions conditionnelles. Ceux-ci seront estimés à l'aide d'un modèle de régression multiple.

La figure 5.14 permet de distinguer approximativement trois saisons dans la série des apports :

Saison 1 : semaine 1 à 15

Saison 2 : semaine 16 à 24

Saison 3 : semaine 25 à 52

Pour refléter cette saisonnalité, nous assignons à chaque semaine  $t$  de l'échantillon ( $t = 1, \dots, 1716$ ) un numéro de saison  $\kappa(t) \in \{1, 2, 3\}$ . Introduisons les variables binaires :

$$R_{kt} = \begin{cases} 1 & \text{si } \kappa(t) = k \\ 0 & \text{sinon} \end{cases}$$

Comme  $\sum_{k=1}^3 R_{kt} = 1$ , nous n'avons besoin que de deux de ces trois variables pour retrouver la 3<sup>ième</sup>. Nous considérons ici la saison 3 comme la référence. Le modèle d'estimation s'écrit :

$$\ln Q_t = a_0 + a_1 \ln Q_{t-1} + \sum_{k=1}^2 \gamma_k R_{kt} + \alpha_t, \quad \alpha_t \sim N(0, \sigma^2), \quad t = 1, \dots, 1716. \quad (5.32)$$

Dans ce modèle, la variance  $\sigma^2(Q_{t-1})$  est stationnaire. Elle est estimée par la variance non biaisée des résidus  $s_\alpha^2 = \frac{\sum \text{résidus}^2}{1716-2}$ . Pour tout  $Q_{t-1}$ , la moyenne conditionnelle  $\mu_t(Q_{t-1})$  est estimée par  $\hat{a}_0 + \hat{a}_1 \ln Q_{t-1} + \sum_{k=1}^2 \hat{\gamma}_k R_{kt}$ ,  $t = 1, \dots, 52$ . Nous prenons les apports de la semaine 52 comme variable hydrologique à la semaine 1, étant donné qu'il s'agit d'un cycle.

La table 5.3 montre les résultats de l'analyse de régression. Elle indique les estimations ponctuelles, la valeur  $p$  (bilatérale) et les limites inférieures et supérieures des intervalles de confiance (respectivement LIIC et LSIC) à 95% de chacun des paramètres de la régression.

Tous ces paramètres sont statistiquement différents de zéro (valeur de  $p$  de 0.000). Le paramètre  $a_1$  étant statistiquement non nul, la forte présomption d'autocorrélation mentionnée plus haut semble se confirmer.

Les résultats indiquent aussi que la régression est significative ( $p = 0.000$ ). Cependant, la signification des variables muettes ne doit pas être testée individuellement. La pertinence de leur introduction dans le modèle peut être validée en comparant deux modèles de régression : avec et sans saisonnalité, par un test de Fischer. La valeur  $p$  de la statistique de ce test ( $F$ ) est de 0.000.

Le coefficient de détermination est de 0.823 pour le modèle avec saisonnalité et de 0.801 pour le modèle réduit. Donc, l'introduction des variables de saison contribuent à expli-



quer à peu près 2.2% de la variation des apports de plus par rapport au modèle réduit. Il faut aussi remarquer que la prise en compte de la saisonnalité réduit significativement l'intervalle de confiance du coefficient d'autocorrélation  $a_1$ .

TABLE 5.3 – Résultats de l'analyse de régression

Modèle avec saisonnalité				
Paramètres	Est. ponct.	Valeur $p$	LIIC	LSIC
$a_0$	1.77	0.000	1.560	1.980
$a_1$	0.752	0.000	0.723	0.781
$\gamma_1$	-0.119	0.000	-0.14 8	-0.089
$\gamma_2$	0.202	0.000	0.170	0.234
$R^2$	0.823	–	–	–
Signification de la régression	–	0.000	–	–
Modèle sans saisonnalité				
Paramètres	Est. ponct.	Valeur $p$	LIIC	LSIC
$a_0$	0.751	0.000	0. 600	0.903
$a_1$	0.895	0.000	0.874	0.912
$R^2$	0.801	–	–	–
Signification de la régression	–	0.000	–	–
Comparaison des deux modèles				
Changement dans $R^2$	0.022	–	–	–
Signification de la statistique $F$	–	0.000	–	–

Revenons maintenant à l'estimation des probabilités conditionnelles. À la semaine  $t$ , supposons que les valeurs possibles des apports soient divisées en  $l_t$  classes dont  $q_t^1, \dots, q_t^{l_t}$  en sont respectivement des valeurs particulières, comme par exemple la limite supérieure, le centre de classe ou la médiane. Soit  $F_t(Q_t = q_t^i | Q_{t-1})$  la fonction de répartition de la loi log-normale de  $(Q_t = q_t^i | Q_{t-1})$  de paramètres  $\mu_t(Q_{t-1})$  et  $\sigma^2(Q_{t-1})$  tels que estimés par le modèle de régression (5.4). Pour chaque valeur  $Q_t = q_t$ , nous prenons  $P(Q_t = q_t^1 | Q_{t-1}) = F_t(q_t^1)$ ,  $P(Q_t = q_t^i | Q_{t-1}) = F_t(q_t^i) - F_t(q_t^{i-1})$ ,  $i = 2, \dots, l_t - 1$ , et  $P(Q_t = q_t^{l_t} | Q_{t-1}) = 1 - F_t(q_t^{l_t-1})$ . Donc, la probabilité (conditionnelle) de chaque valeur est estimée par celle de sa classe.

## 5.5 Cadre expérimental

Une approche en deux phases (estimation des fonctions de valeur et simulation de la politique de gestion) a été adoptée pour tester la méthodologie d'approximation simpliciale sur le système de RTA. À chaque semaine  $t$ , nous avons estimé la fonction de valeur telle que présentée dans les sections 5.3 et 5.4 ci-dessus. Les distributions de

probabilités ont été construites à l'aide des apports historiques totaux montrés dans la figure 5.14 plus haut.

La valeur finale de l'eau  $V_{53}$  a été calculée par « rodage ». Elle a tout d'abord été fixée à une valeur nulle. Les fonctions de valeur ont été estimées sur 52 semaines trois fois de suite. À la fin du rodage, nous avons choisi  $V_{53} = V_1$  puisqu'il s'agit d'un cycle. Avec cette valeur terminale, les fonctions de valeur ont été construites de nouveau sur 52 semaines.<sup>1</sup>

Ensuite, la politique de gestion a été simulée à l'aide de 25 autres années d'apports historiques totaux et les fonctions de valeur construites à la phase précédente. Pour un état observé  $(v_t, Q_t)$ , la politique optimale est donnée par  $[u_t, v_{t+1}, y_t] \in \text{Argmax}\{\hat{G}_t(v_t, Q_t)\}$  en résolvant le programme (5.12-5.30).

Un stock initial  $v_1$  a été choisi, correspondant à la capacité maximale du premier réservoir et à peu près 98% de celle du second. Ces choix (apports et  $v_1$ ) ont été faits aux fins de comparaison des résultats avec ceux obtenus avec un modèle de PDS construit par RTA (dans ce qui suit, ce modèle sera appelé « modèle RTA »).

Le modèle RTA utilise une grille régulière à chaque période de  $10 \times 30 \times 7$  points (ces chiffres correspondent respectivement au nombre de points de discrétisation des niveaux du premier et du deuxième réservoirs, et des apports) pour évaluer la fonction de valeur. Dans ce modèle, les apports de la semaine  $t - 1$  sont aussi la variable hydrologique de la semaine  $t$ .

Il existe deux grandes différences entre les phases d'estimation et de simulation. Dans la première, les fonctions de valeur de l'eau sont construites par la PDS par induction à rebours en utilisant des distributions de probabilité sur les apports. Dans la simulation, les apports sont connus à chaque semaine et la politique d'opération est calculée par la programmation dynamique par « induction en avant ». À la première semaine, le programme (5.12-5.30) est résolu pour l'état  $(v_1, Q_1)$ . Ensuite, le programme est résolu pour  $(v_2, Q_2)$ , où  $v_2$  est le stock final optimal obtenu à la première semaine. Ce procédé est répété jusqu'à la semaine 1300.

### 5.5.1 Résultats et analyses

Nous avons tout d'abord estimé les fonctions de valeur sans variable hydrologique, i.e.  $\hat{V}_t(v_t) = E_{Q_t}[\hat{G}_t(v_t, Q_t)] = \sum_{i=1}^I P(Q_t = q_t^i) \hat{G}_t(v_t, Q_t)$ . Nous appellerons ce modèle

1. C'est ce que nous avons fait aussi dans la section 5.3.1.

« modèle sans variable hydrologique » ou « MSVH ». Ensuite, la variable hydrologique  $Q_{t-1}$  a été introduite dans le modèle. Dans ce cas, l'estimation de la fonction de valeur est donnée par (5.4). Ce modèle sera appelé « modèle avec variable hydrologique » ou « MAVH ». Les résultats sont indiqués dans les sections 5.5.1.1 et 5.5.1.2 respectivement.

Les débits turbinés ont été discrétisés selon la stratégie  $ST^3$  décrite dans la section 5.3.1, i.e. celle avec le moins de points.

Après avoir construit les fonctions de valeur de l'eau, la politique de gestion a été simulée comme nous l'avons indiqué plus haut. Aux fins de comparaison, nous avons estimé, pour chaque centrale, la puissance correspondant à la politique optimale, prescrite par le programme (5.12-5.30), à l'aide des mêmes polynômes utilisés par le modèle RTA.

### 5.5.1.1 Modèle sans variable hydrologique

Les apports ont tout d'abord été discrétisés selon les valeurs  $\mu_t$  et  $s_t^2$  de la loi log-normale :  $\mu_t - 3s_t, \mu_t - 2s_t, \mu_t, \mu_t + 2s_t, \mu_t + 3s_t$ . Les paramètres  $\mu_t$  et  $s_t^2$  ont été estimés respectivement par les moyennes et les variances des apports historiques totaux de la phase d'estimation. Les probabilités des apports ont été calculées comme nous l'avons discuté à la fin de la section 5.4 à l'aide de distributions non conditionnelles, donc sans le modèle de régression.

Pour tester la sensibilité de la solution optimale par rapport au nombre de points de discrétisation des apports, nous avons simulé la politique optimale en faisant varier le nombre de points de discrétisation des apports. À partir de sept points, la valeur optimale n'a changé que de manière insignifiante.

Pour chaque centrale, nous avons calculé l'efficacité moyenne de la politique obtenue avec les deux modèles (RTA et MSVH). Celle-ci a été définie comme le ratio de la puissance moyenne aux débits totaux (turbinés et délestés). Les résultats sont montrés dans la table 5.4. Pour chaque semaine, l'algorithme de rehaussement (voir la section 3.3.3) a été exécuté pour générer les simplexes initiaux. La grille initiale a été raffinée selon une tolérance d'erreur  $\eta = 0.8$ . Globalement, l'efficacité moyenne de la politique de gestion prescrite par les deux modèles est similaire (MSVH et RTA).

Nous avons essayé de réduire la taille de la grille des stocks en générant les simplexes initiaux avec l'algorithme CFK (pour un rappel, voir la section 3.3.3). Nous appelons ce modèle  $MSVH^{CFK}$ . L'efficacité moyenne globale ( $MSVH^{CFK}$ ) est comparable à celles des deux autres modèles.

Cependant, à chaque semaine, la taille moyenne de la grille des stocks avec le modèle  $MSVH^{CFK}$  était de 8 approximativement et la taille maximale de 9 contre 13 et 18 respectivement pour le MSVH. Avec le MSVH, le temps de résolution (construction des fonctions de valeur) était d'environ 46.0609 s. Avec le modèle  $MSVH^{CFK}$ , ce temps a été réduit à 24.4793 s. Les efforts de calcul ont donc été divisés par environ 1.88.

TABLE 5.4 – Comparaison entre le modèle RTA, MSVH et le modèle  $MSVH^{CFK}$  : efficacité moyenne de la politique de gestion et nombre de points de discrétisation des stocks par semaine

Modèle	Efficacité moyenne en $MW/m^3/s$				
	CCD	CCS	CIM	CSH	Moyenne globale
RTA	0.278	0.295	0.252	0.566	0.348
MSVH	0.261	0.308	0.252	0.563	0.346
$MSVH^{CFK}$	0.256	0.306	0.249	0.568	0.345
Modèle	Nombre de points de discrétisation des stocks finaux par semaine				
	Min	Max	Moyenne	–	–
RTA	300	300	300	–	–
MSVH	13	18	14	–	–
$MSVH^{CFK}$	8	9	8	–	–

Nous avons ensuite testé la méthode avec des tolérances d'erreur plus faibles afin de comparer le gain en efficacité avec les efforts de calcul. La table 5.5 montre des statistiques sur le nombre de points de grille des stocks finaux par période, les efficacités moyennes globales et le temps de calcul. Le gain en efficacité est très faible lorsqu'on diminue la tolérance d'erreur, i.e. lorsqu'on augmente le nombres de points de discrétisation des stocks finaux. Le gain en puissance moyenne est d'environ 0.067% seulement, tandis que le temps de calcul augmente « considérablement ».

La figure 5.15 montre des exemples de grille des stocks ( $\eta = 0.5$ ) obtenues avec le modèle  $MSVH^{CFK}$  à trois périodes différentes ainsi qu'une grille régulière de dimension  $10 \times 30$  utilisé dans le modèle RTA. On voit que le raffinement de la grille est différente entre les trois périodes étant donné que l'incurvation de la fonction de valeur peut être différente d'une période à l'autre.

TABLE 5.5 – Compromis entre l’efficacité globale moyenne et le temps de calcul pour les modèles sans variable hydrologique

Modèle	Efficacité moyenne en $MW/m^3/s$			
	$\eta = 0.5$	$\eta = 0.1$	$\eta = 0.05$	$\eta = 0.001$
MSVH	0.346	0.346	0.347	0.347
$MSVH^{CFK}$	0.346	0.347	0.347	0.347
Modèle	Temps de calcul en secondes			
	$\eta = 0.5$	$\eta = 0.1$	$\eta = 0.05$	$\eta = 0.001$
MSVH	63.896	241.054	476.533	3 230.600
$MSVH^{CFK}$	29.862	86.000	177.932	2 553.900
Modèle	# moyen de points de discrétisation des stocks finaux par semaine			
	$\eta = 0.5$	$\eta = 0.1$	$\eta = 0.05$	$\eta = 0.001$
MSVH	17.865	61.558	110.058	486.962
$MSVH^{CFK}$	9.596	24.865	46.962	421.038

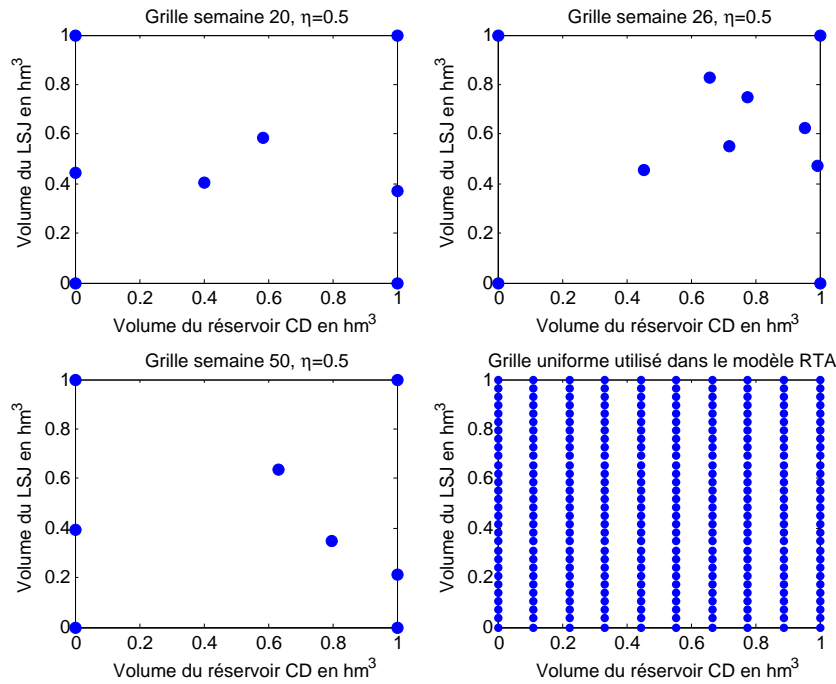


FIGURE 5.15 – Trois exemples de grille à trois périodes différentes et un exemple de grille régulière

### 5.5.1.2 Modèle avec variable hydrologique

Dans le modèle avec variable hydrologique (MAVH), à chaque semaine, une grille de stocks finaux différente a été construite pour chaque valeur de la variable hydrologique. Ceci est justifié par le fait que tout d’abord, à une semaine donnée, chaque valeur de la variable hydrologique peut conduire à des états d’information différents à la semaine

suivante. De plus, pour raffiner la grille des stocks finaux, on utilise les évaluations de la fonction de valeur aux sommets des simplexes (pour un rappel, voir la section 3.3.4). Or, ces évaluations dépendent des valeurs de la variable hydrologique.

Les apports ont été discrétisés en plusieurs classes d'amplitudes différentes de sorte que la probabilité de transition d'une classe à la semaine  $t$  à une autre classe à la semaine  $t + 1$  soit différente de zéro. Les centres de classe ont été choisis comme les points de discrétisation. Des approches semblables sont discutées dans Karamouz et Vasiliadis (1992).

Dans la « logique » du modèle de régression (5.4), nous avons construit une discrétisation de base (pour la semaine 25 à 52) en nous référant à l'étendue des apports historiques totaux (utilisés dans la phase d'estimation). Pour les autres semaines, la discrétisation de base a été corrigée par les facteurs saisonniers estimés (voir la table 5.3).

Puisque  $Q_{t-1}$  est la variable hydrologique à la semaine  $t$ , celle-ci a donc la même discrétisation que les apports à la semaine  $t - 1$ .

Comme précédemment, après avoir estimé les fonctions de valeur de l'eau, nous avons simulé la politique de gestion avec les 25 années d'apports historiques. Aux fins de comparaison, nous avons résolu les modèles MSVH et MAVH avec sept valeurs de discrétisation des apports. Dans les deux cas, les simplexes initiaux ont été produits par l'algorithme CFK. Des statistiques sur l'efficacité globale, le nombre de points de grille des stocks finaux ainsi que les temps de calcul pour différentes tolérances d'erreur sont indiqués dans la table 5.6.

Globalement, les efficacités moyennes sont comparables avec un léger avantage pour le MSVH. L'inclusion de la variable hydrologique ne semble pas améliorer l'efficacité globale alors que la complexité des efforts de calcul augmente (estimation des fonctions de valeur). Plusieurs facteurs pourraient expliquer ce fait.

Premièrement, peut être que  $Q_{t-1}$  ne permet pas d'intégrer correctement la persistance dans le modèle et y rajoute du bruit. Deuxièmement, il se peut que le modèle de régression utilisé ne permette pas de représenter correctement la distribution conditionnelle des apports. Peut-être qu'il manque d'autres variables pertinentes. Ce modèle suppose que la variance des apports est stationnaire, peut-être qu'un modèle de régression généralisée hétéroscédastique serait plus approprié.

Étant donné que la modélisation des séries hydrologiques n'est pas l'objet de cette thèse,

nous ne nous pencherons pas sur cette question. Toutefois, mentionnons que l'étude de Tejada-Guibert *et al.* (1995) a montré que la « valeur » de l'utilisation de modèles de prévision des apports sophistiqués dans la PDS dépend de plusieurs facteurs, dont les caractéristiques du système et les objectifs d'opération. Cette conclusion a été tirée à partir de l'analyse des valeurs optimales réelles (vérifiées par simulation) obtenues en utilisant plusieurs modèles de PDS avec et sans variable(s) hydrologique(s).

Observons qu'avec le MAVH, on obtiendrait un gain en efficacité de  $0.001 \text{ MW}/\text{m}^3/\text{s}$  par rapport au modèle RTA. Cette différence correspond approximativement à un gain moyen annuel de 315 360 \$. En effet, le débit moyen annuel du système est de  $900 \text{ m}^3/\text{s}$  et le prix du  $MWh$  est de 40 \$. Sachant qu'il y a 8 760 heures dans une année, le gain monétaire est donc :

$$\begin{aligned} GA &= 0.001 \frac{MW}{m^3/s} \times \frac{900 \text{ m}^3/s}{an} = 0.9 \frac{MW}{an} \times \frac{8760 \text{ h}}{an} \\ &= 7884 \text{ MWh} \times 40 \text{ \$/MWh} = 315 \text{ 360 \$} \end{aligned}$$

Notons aussi que le MSVH permettrait d'obtenir un gain économique double, car le gain en efficacité par rapport au modèle RTA est de l'ordre de  $0.002 \text{ MW}/\text{m}^3/\text{s}$ .

TABLE 5.6 – Comparaison entre les modèles MAVH et MSVH : efficacité moyenne de la politique de gestion, nombre moyen de points de discrétisation des stocks par semaine et temps de résolution

Modèle	Efficacité moyenne en $MW/m^3/s$			
	$\eta = 0.8$	$\eta = 0.5$	$\eta = 0.1$	
MSVH	0.350	0.350	0.351	–
MAVH	0.349	0.349	0.349	–
RTA	–	–	–	0.348
Modèle	# moyen de points de discrétisation des stocks par semaine			
	$\eta = 0.8$	$\eta = 0.5$	$\eta = 0.1$	
MSVH	8.288	9.731	25.442	
MAVH	59.000	68.327	188.538	
Modèle	Temps de résolution en secondes			
	$\eta = 0.8$	$\eta = 0.5$	$\eta = 0.1$	
MSVH	26.062	32.497	99.453	
MAVH	175.032	200.144	662.166	

Les figures 5.16 et 5.17 montrent les trajectoires optimales des réservoirs prescrites par les modèles RTA et MSVH sur les 25 années de simulation. La politique de gestion est comparable pour le LSJ dont le niveau est baissé en début d'année en prévision des crues printanières. Ensuite, le réservoir est rempli progressivement. Après la saison estivale, où la trajectoire doit passer dans un corridor serré, le niveau est remonté. Sur

le reste de l'année, le niveau fluctue en fonction de l'intensité des apports.

Dans le cas du réservoir Chute-Du-Diable, avec les deux modèles, le niveau est baissé en prévision de la fonte de la neige au printemps comme dans le cas du LSJ. Après le printemps, son niveau fluctue beaucoup plus dans le MSVH.

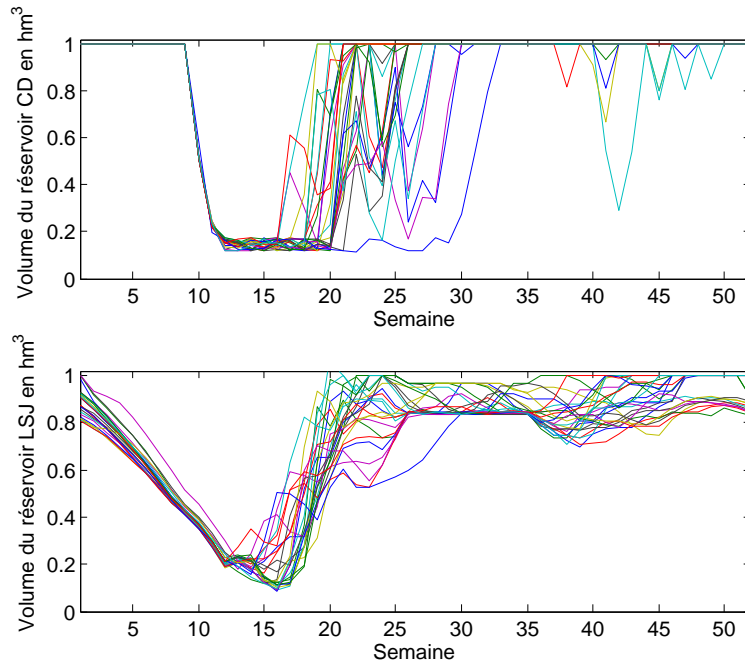


FIGURE 5.16 – Trajectoire hebdomadaire optimale des réservoirs prescrite par le modèle RTA sur les 25 années de simulation

La figure 5.18 montre la trajectoire hebdomadaire moyenne des réservoirs prescrites par les modèles RTA et MSVH. À l'exception du printemps, le modèle RTA gère le réservoir Chute-Du-Diable à un niveau plus élevé comparativement au MSVH. Cette tendance est inversée dans le cas du LSJ.

Les résultats du MAVH, comme ceux présentés dans la section précédente, montrent qu'agrandir la grille des stocks finaux, c'est-à-dire évaluer la fonction de valeur un grand nombre de fois à chaque période, ne permet pas pour autant d'améliorer la politique de gestion. En effet, en diminuant la tolérance d'erreur, l'efficacité moyenne reste pratiquement inchangée (table 5.6). Donc, pour avoir une bonne approximation, il suffit de choisir quelques points « bien positionnés » auxquels la fonction de valeur est évaluée, et interpoler la politique pour des points qui ne sont pas dans la grille.



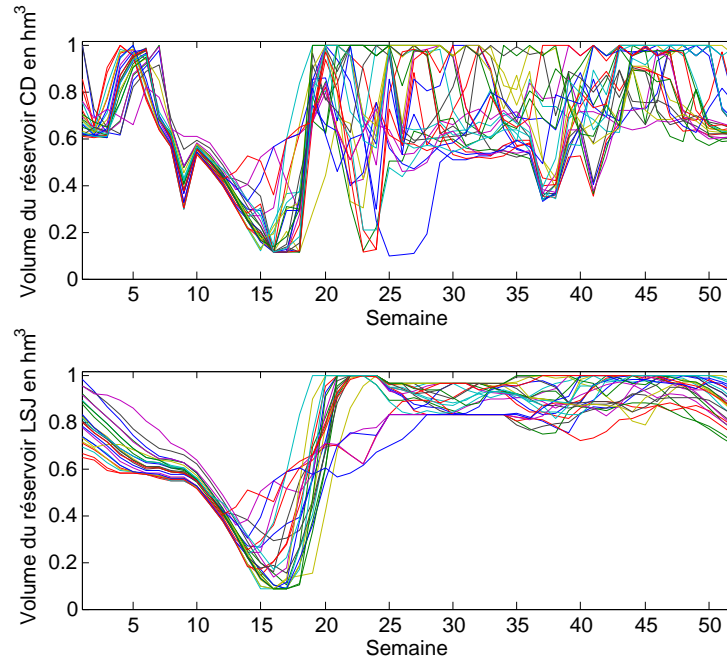


FIGURE 5.17 – Trajectoire hebdomadaire optimale des réservoirs prescrite par le MSVH sur les 25 années de simulation

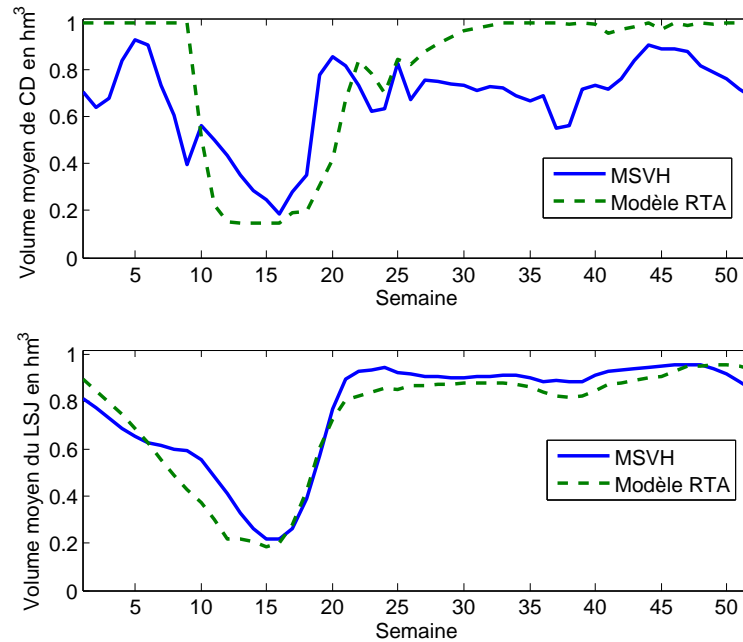


FIGURE 5.18 – Trajectoire optimale moyennes des réservoirs prescrite par les modèles RTA et MSVH

Bien que nous ayons observé qu'au delà de sept valeurs de la variable hydrologique, les résultats n'ont pas été améliorés, à titre d'illustration, la figure 5.19 montre l'évolution du temps de calcul (construction des fonctions de valeur) en fonction du nombre de valeurs de discrétisation. Le seuil utilisé sur la tolérance d'erreur était  $\eta = 0.8$ . Observons que la complexité de calcul augmente de manière plus que proportionnelle avec le nombre de points de discrétisation de la variable hydrologique.

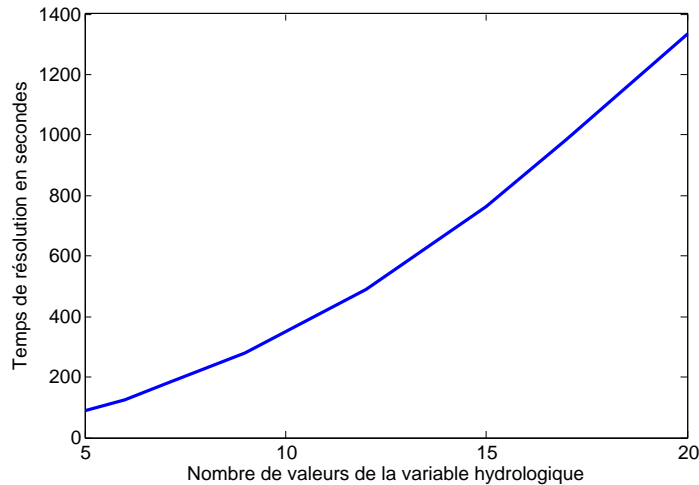


FIGURE 5.19 – Évolution du temps de calcul en fonction du nombre de valeurs de la variable hydrologique

## 5.6 Conclusion

Ce chapitre a présenté une application de la méthode de programmation dynamique approximative développée dans cette thèse au contexte de gestion à moyen terme d'une partie du système hydroélectrique de l'entreprise Rio Tinto Alcan. Après avoir décrit le système étudié et discuté d'une adaptation de la méthodologie au contexte d'étude, nous avons testé la méthode en suivant une procédure en deux étapes.

Tout d'abord, les fonctions de valeur de l'eau ont été estimées par la programmation linéaire généralisée. Trente-trois années d'apports naturels historiques ont été modélisés par une régression multiple, ce qui a permis de construire, pour chaque période, la distribution de probabilité conditionnelle des apports.

Ensuite la politique de gestion a été simulée sur un horizon de 25 années. L'efficacité moyenne globale (puissance totale moyenne par unités de débits) est comparable à celle obtenue avec un modèle de programmation dynamique stochastique classique. Cependant, la complexité de calcul a été réduite significativement avec notre modèle.

En effet, à chaque période, la fonction de valeur a été évaluée beaucoup moins souvent que dans le modèle classique où, à chaque semaine, elle a été évaluée sur une grille uniforme très fine.

Les résultats montrent que pour obtenir une « bonne approximation », il n'est pas nécessaire d'avoir une grille très dense, à chaque période, il suffit de choisir « convenablement » quelques points de discrétisation.



# Conclusion générale, limites et perspectives de recherche future

Cette thèse a étudié le problème de la planification à moyen terme des systèmes multi-réservoirs par l'optimisation stochastique. Ce problème est difficile à cause principalement de la nature incertaine des apports naturels aux réservoirs qui fluctuent dans l'espace et le temps. Ceci rend inefficaces les méthodes déterministes qui ne permettent pas de représenter la dynamique information-décision.

Plusieurs méthodes stochastiques ont été proposées pour ces problèmes, dont l'agrégation de scénarios et la programmation dynamique stochastique approximative. Celle-là est basée sur un algorithme (*progressive hedging*) dont le taux de convergence dépend d'un paramètre de pénalisation. Il a été constaté empiriquement qu'un mauvais choix de valeur de ce paramètre peut nuire à la convergence numérique de l'algorithme.

Toutefois, il n'existe aucune indication théorique sur le choix d'une valeur appropriée pour ce paramètre. Sa valeur ne peut être mise à jour que de manière heuristique. L'approche classique consiste à utiliser une valeur fixe (choisie empiriquement). D'autres approches ont proposé une mise à jour selon une trajectoire fixe : croissante ou décroissante.

Nous avons étudié une méthode de mise à jour dynamique, i.e. qu'à certaines itérations de l'algorithme, nous permettons à la valeur du paramètre de diminuer et à d'autres itérations d'augmenter. La trajectoire dépend de l'« état de la solution », mesuré à l'aide de deux indicateurs de performance.

Les essais numériques réalisés avec différentes configurations de réservoirs et différents scénarios d'apports et de demande ont montré que notre approche donne de meilleurs résultats que l'approche classique, à la fois en termes de taux de convergence et de nombre d'itérations.

Notre étude a considéré un seul paramètre de pénalisation pour toutes les contraintes

de non anticipativité. Étant donné que les limites des réservoirs peuvent être différentes, il serait peut-être avantageux d'utiliser un paramètre différent pour chaque stock.

Nous avons testé notre méthode dans un contexte d'arbre d'événements donné. Des travaux futurs pourraient inclure une étape de construction d'arbres d'événements « représentatifs » des processus stochastiques.

Une approche courante construit un arbre d'événements très grands au début et le réduit itérativement avant de l'appliquer au problème à résoudre (e.g Dupačová *et al.*, 2003; Barro *et al.*, 2008; Heitsch et Römisch, 2003). Il existe plusieurs techniques de réduction, mais essentiellement, l'arbre est réduit tant que la distribution de probabilité sous-jacente est « proche » de la distribution de probabilité initiale selon une métrique donnée. Donc, deux « tâches », possiblement coûteuses en termes d'efforts de calcul, sont réalisées : construction de l'arbre et sa réduction.

Cependant, nous nous demandons si une stratégie inverse ne serait pas meilleure. Au lieu de construire un arbre « gigantesque » et le réduire ensuite, ne pourrait-on pas commencer par construire un arbre petit et l'enrichir progressivement jusqu'à atteindre un degré de précision fixé qui pourrait, par exemple, être basé sur une certaine métrique ? Ceci peut l'objet d'un sujet de recherche futur.

La deuxième partie de la thèse s'est portée sur l'application de la programmation dynamique stochastique à la gestion des réservoirs. Plusieurs méthodes d'approximation de la fonction de valeur existent. Cependant, à notre connaissance, il n'y a pas beaucoup d'applications aux problèmes de réservoirs avec contrôle de l'erreur d'approximation. Nous avons développé une approche basée sur le partitionnement de l'espace des stocks en simplexes. La fonction de valeur est évaluée aux sommets de ces simplexes, qui forment une grille irrégulière, et interpolée ailleurs. De plus, nous avons exploité l'hypothèse de corrélation spatiale parfaite (entre les apports naturels) pour réduire la complexité du calcul de l'espérance de cette fonction. Au lieu d'utiliser des méthodes d'intégration numérique, nous obtenons des formes analytiques de l'espérance.

Les problèmes de réservoirs sont généralement non convexes à cause principalement des fonctions de production qui sont généralement non concaves et non linéaires. Nous avons utilisé une approche d'optimisation par la programmation linéaire généralisée pour contourner cette difficulté, ce qui permet l'utilisation des « solveurs » de programmation linéaire répandus.

La méthodologie proposée a été testée à l'aide de plusieurs séries de problèmes simulés

et dans un contexte industriel. Dans l'étude de cas, nous avons obtenu des efficacités (puissance moyenne par unité de débits) comparables à celles d'un modèle de programmation dynamique stochastique classique (évaluation de la fonction de valeur sur une grille régulière). Cependant, dans notre modèle, les efforts de calcul ont été réduits considérablement. En effet, à chaque période, la fonction de valeur a été approximée sur une grille irrégulière, beaucoup moins dense qu'une grille uniforme.

Nous croyons que la méthode pourrait être améliorée, notamment quant à l'inclusion d'une variable hydrologique. À chaque période, pour chaque valeur de la variable hydrologique, la fonction de valeur est évaluée sur une grille différente, construite en divisant un ensemble de simplexes existants. Les efforts de calcul peuvent augmenter si le nombre de réservoirs est élevé et si la variable hydrologique est multidimensionnelle. Il serait peut-être plus efficace de discrétiser conjointement l'espace des états, c'est-à-dire les stocks finaux et la variable hydrologique. Des études futures s'avèrent nécessaires pour analyser ce sujet.

Il pourrait aussi être intéressant de doubler chaque itération de la division de simplexes d'une étape de simulation des apports naturels et de raffiner la grille non seulement en fonction de l'incurvation de la fonction de valeur mais aussi en tenant compte des régions les plus susceptibles d'être visitées par le système.

Finalement, l'hypothèse de corrélation spatiale parfaite sur laquelle repose la simplification du calcul de l'espérance de la fonction de valeur pourrait causer de l'imprécision dans la politique de gestion prescrite par le modèle d'optimisation dans des systèmes où les conditions climatiques sont fortement contrastées.





# Bibliographie

- Ahmed, J. A. et Sarma, A. K. Genetic algorithm for optimal operating policy of a multipurpose reservoir. *Water Resources Management*, 19(2) :145–161, 2005.
- Anctil, F., Rousselle, J., et Lauzon, N. *Hydrologie : cheminements de l'eau*. Presses inter Polytechnique, 2012.
- Barro, D., Canestrelli, E., et Ciurlia, P. Spatial aggregation in scenario tree reduction. *Mathematical and Statistical Methods in Insurance and Finance*, p. 27–34. Springer, 2008.
- Barros, M. T., Tsai, F. T., Yang, S.-l., Lopes, J. E., et Yeh, W. W. Optimization of large-scale hydropower system operations. *Journal of Water Resources Planning and Management*, 129(3) :178–188, 2003.
- Bazaraa, M., Jarvis, J., et Sherali, H. *Linear Programming and Network Flows*. Wiley, New York, 1990.
- Bazaraa, M. S., Sherali, H. D., et Shetty, C. M. *Nonlinear programming : theory and algorithms*. John Wiley & Sons, 2013.
- Bellman, R. *Dynamic Programming*. Princeton University Press, Princeton, 1957.
- Benders, J. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4 :238–252, 1962.
- Bertsekas, D. P. et Tsitsiklis, J. N. Neuro-dynamic programming : an overview. *Proceedings of the 34th IEEE Conference on Decision and Control*, t. 1, p. 560–564. IEEE, 1995.
- Birge, J. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, p. 989–1007, 1985.

- Birge, J. et Louveaux, F. *Introduction to Stochastic Programming*. Springer-Verlag, 1997.
- Blomvall, J. et Lindberg, P. O. A Riccati-based primal interior point solver for multistage stochastic programming. *European Journal of Operational Research*, 143(2) :452–461, 2002.
- Cai, X., McKinney, D. C., et Lasdon, L. S. Solving nonlinear water management models using a combined genetic algorithm and linear programming approach. *Advances in Water Resources*, 24(6) :667–676, 2001.
- Candler, G. V. Finite-Difference Methods for Continuous-Time Dynamic Programming. M. Ramon et S. Andrew, réds., *Computational Methods for the Study of Dynamic Economies*. Oxford University Press, 2001.
- Carpentier, P. L., Gendreau, M., et Bastin, F. Long-term management of a hydroelectric multireservoir system under uncertainty using the progressive hedging algorithm. *Water Resources Research*, 49(5) :2812–2827, 2013.
- Castelletti, A., de Rigo, D., Rizzoli, A. E., Soncini-Sessa, R., et Weber, E. Neurodynamic programming for designing water reservoir network management policies. *Control Engineering Practice*, 15(8) :1031–1038, 2007.
- Castelletti, A., Galelli, S., Restelli, M., et Soncini-Sessa, R. Tree-based reinforcement learning for optimal water reservoir operation. *Water Resources Research*, 46(9), 2010.
- Celeste, A. B. et Billib, M. Evaluation of stochastic reservoir operation optimization models. *Advances in Water Resources*, 32(9) :1429–1443, 2009.
- Cervellera, C., Chen, V. C., et Wen, A. Optimization of a large-scale water reservoir network by stochastic dynamic programming with efficient state space discretization. *European Journal of Operational Research*, 171(3) :1139–1151, 2006.
- Chang, G. W., Aganagic, M., Waight, J. G., Medina, J., Burton, T., Reeves, S., et Christoforidis, M. Experiences with mixed integer linear programming based approaches on short-term hydro scheduling. *IEEE Transactions on Power Systems*, 16(4) :743–749, 2001.
- Chen, L. et Chang, F. Applying a real-coded multi-population genetic algorithm to multi-reservoir operation. *Hydrological Processes*, 21(5) :688–698, 2007.

- Côté, P. *Gestion d'un ensemble d'installations hydroélectriques soumis à une contrainte de demande*. Thèse de doctorat, École Polytechnique de Montréal, 2010.
- Crawley, P. D. et Dandy, G. C. Optimal operation of multiple-reservoir system. *Journal of Water Resources Planning and Management*, 119(1) :1–17, 1993.
- Dickinson, P. J. On the exhaustivity of simplicial partitioning. *Journal of Global Optimization*, p. 1–15, 2013.
- Dos Santos, M., Da Silva, E., Finardi, E., et Gonçalves, R. Practical aspects in solving the medium-term operation planning problem of hydrothermal power systems by using the progressive hedging method. *International Journal of Electrical Power & Energy Systems*, 31(9) :546–552, 2009.
- Dupačová, J., Gröwe-Kuska, N., et Römisch, W. Scenario reduction in stochastic programming - An approach using probability metrics. *Mathematical Programming*, 95(3) :493–511, 2003.
- Foufoula-Georgiou, E. et Kitanidis, P. K. Gradient dynamic programming for stochastic optimal control of multidimensional water resources systems. *Water Resources Research*, 24(8) :1345–1359, 1988.
- Gal, S. The parameter iteration method in dynamic programming. *Management Science*, 35(6) :675–684, 1989.
- Gonçalves, R., Finardi, E., et da Silva, E. Exploring the progressive hedging characteristics in the solution of the medium-term operation planning problem. *Proc. 17th Power Systems Computation Conference*. PSCC, Stockholm, Sweden, 2011.
- Goor, Q., Kelman, R., et Tilmant, A. Optimal multipurpose-multireservoir operation model with variable productivity of hydropower plants. *Journal of Water Resources Planning and Management*, 137(3) :258–267, 2010.
- Grüne, L. et Semmler, W. Using dynamic programming with adaptive grid scheme for optimal control problems in economics. *Journal of Economic Dynamics and Control*, 28(12) :2427–2456, 2004.
- Hall, W. A., Butcher, W. S., et Esogbue, A. Optimization of the operation of a multiple-purpose reservoir by dynamic programming. *Water Resources Research*, 4(3) :471–477, 1968.

- Hammadia, A. *Contributions à l'optimisation, en temps réel et à court terme, des ressources hydroélectriques d'une rivière*. Thèse de doctorat, École Polytechnique de Montréal, 2001.
- Heitsch, H. et Römisich, W. Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, 24(2-3) :187–206, 2003.
- Hiriart-Urruty, J.-B. et Lemaréchal, C. *Fundamentals of Convex Analysis*. Springer, 2001.
- Jiekang, W., Jianquan, Z., Guotong, C., et Hongliang, Z. A hybrid method for optimal scheduling of short-term electric power generation of cascaded hydroelectric plants based on particle swarm optimization and chance-constrained programming. *IEEE Transactions on Power Systems*, 23(4) :1570–1579, 2008.
- Johnson, S. A., Stedinger, J. R., Shoemaker, C. A., Li, Y., et Tejada-Guibert, J. A. Numerical solution of continuous-state dynamic programs using linear and spline interpolation. *Operations Research*, 41(3) :484–500, 1993.
- Jothiprakash, V. et Shanthi, G. Single reservoir operating policies using genetic algorithm. *Water Resources Management*, 20(6) :917–929, 2006.
- Karamouz, M. et Houck, M. H. Annual and monthly reservoir operating rules generated by deterministic optimization. *Water Resources Research*, 18(5) :1337–1344, 1982.
- Karamouz, M. et Vasiliadis, H. V. Bayesian stochastic optimization of reservoir operation using uncertain forecasts. *Water Resources Research*, 28(5) :1221–1232, 1992.
- Kaut, M. et Wallace, S. Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, 3(2) :257–271, 2007.
- Kelman, J., Stedinger, J. R., Cooper, L. A., Hsu, E., et Yuan, S.-Q. Sampling stochastic dynamic programming applied to reservoir operation. *Water Resources Research*, 26(3) :447–454, 1990.
- Kim, Y.-O., Eum, H.-I., Lee, E.-G., et Ko, I. H. Optimizing operational policies of a Korean multireservoir system using sampling stochastic dynamic programming with ensemble streamflow prediction. *Journal of Water Resources Planning and Management*, 133(1) :4–14, 2007.

- Krau, S., Émiel, G., et Merleau, J. Une méthode d'échantillonnage adaptatif de l'espace des états pour la programmation dynamique appliquée à la gestion de systèmes hydriques. Séminaire de présentation, CIRRELT, Montréal, Janvier, 2014.
- Labadie, J. W. Optimal operation of multireservoir systems : State-of-the-art review. *Journal of Water Resources Planning and Management*, 130(2) :93–111, 2004.
- Lamond, B. F. et Lang, P. Stochastic optimization of multi-reservoir systems with power plants and spillways. *WIT Transactions on Ecology and the Environment*, 104 :31–40, 2007.
- Lee, J.-H. et Labadie, J. W. Stochastic optimization of multireservoir systems via reinforcement learning. *Water Resources Research*, 43(11) :W11408, 2007.
- Li, X.-Z., Xu, L.-Z., et Chen, Y.-G. Implicit stochastic optimization with data mining for reservoir system operation. *International Conference on Machine Learning and Cybernetics*, t. 5, p. 2410–2415. IEEE, Qingdao, 2010.
- Little, J. The use of storage water in a hydroelectric system. *Operations Research*, 3 :187–197, 1955.
- Moore, D. W. *Simplicial mesh generation with applications*. Thèse de doctorat, Cornell University, 1992.
- Mousavi, H. et Ramamurthy, A. Optimal design of multi-reservoir systems for water supply. *Advances in Water Resources*, 23(6) :613–624, 2000.
- Mulvey, J. et Vladimirov, H. Applying the progressive hedging algorithm to stochastic generalized networks. *Annals of Operations Research*, 31(1) :399–424, 1991.
- Munos, R. et Moore, A. Variable resolution discretization in optimal control. *Machine Learning*, 49(2-3) :291–323, 2002.
- Needham, J. T., Watkins Jr, D. W., Lund, J. R., et Nanda, S. Linear programming for flood control in the Iowa and Des Moines rivers. *Journal of Water Resources Planning and Management*, 126(3) :118–127, 2000.
- Ouarda, T. et Labadie, J. Chance-constrained optimal control for multireservoir system optimization and risk analysis. *Stochastic Environmental Research and Risk Assessment*, 15(3) :185–204, 2001.

- Paulavičius, R. et Žilinskas, J. *Simplicial Global Optimization*. Springer, New York, 2013.
- Pennanen, T. et Kallio, M. A splitting method for stochastic programs. *Stochastic Programming E-Print Series*, 2006.
- Pereira, M. et Pinto, L. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(1) :359–375, 1991.
- Pereira, M. V. F. et Pinto, L. Stochastic optimization of a multireservoir hydroelectric system : A decomposition approach. *Water Resources Research*, 21(6) :779–792, 1985.
- Pérez-Díaz, J. I., Wilhelmi, J. R., et Arévalo, L. A. Optimal short-term operation schedule of a hydropower plant in a competitive electricity market. *Energy Conversion and Management*, 51(12) :2955–2966, 2010.
- Philbrick, C. R. P. et Kitanidis, P. K. Limitations of deterministic optimization applied to reservoir operations. *Journal of Water Resources Planning and Management*, 125(3) :135–142, 1999.
- Rani, D. et Moreira, M. M. Simulation–optimization modeling : a survey and potential application in reservoir systems operation. *Water Resources Management*, 24(6) :1107–1138, 2010.
- Reis, F., Carvalho, P., et Ferreira, L. Reinforcement scheduling convergence in power systems transmission planning. *IEEE Transactions on Power Systems*, 20(2) :1151–1157, 2005.
- Reis, L., Bessler, F., Walters, G., et Savic, D. Water supply reservoir operation by combined genetic algorithm–linear programming (GA-LP) approach. *Water Resources Management*, 20(2) :227–255, 2006.
- Rockafellar, R. T. et Wets, R. J. B. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, p. 119–147, 1991.
- Rust, J. Numerical dynamic programming in economics. *Handbook of Computational Economics*, 1 :619–729, 1996.
- Ruszczynski, A. Some advances in decomposition methods for stochastic linear programming. *Annals of Operations Research*, 85 :153–172, 1999.

- Salinger, D. H. et Rockafellar, R. T. Dynamic splitting : An algorithm for deterministic and stochastic multiperiod optimization. Working paper, Departement of Mathematics, University of Washington, Seattle, 2003.
- Seifi, A. et Hipel, K. W. Interior-point method for reservoir operation with stochastic inflows. *Journal of Water Resources Planning and Management*, 127(1) :48–57, 2001.
- Sen, S. et Higle, J. L. An introductory tutorial on stochastic linear programming models. *Interfaces*, 29(2) :33–61, 1999.
- Shapiro, J. F. *Mathematical Programming : Structures and Algorithms*. Wiley New York, 1979.
- Shawwash, Z. K., Siu, T. K., et Russell, S. D. The BC Hydro short term hydro scheduling optimization model. *IEEE Transactions on Power Systems*, 15(3) :1125–1131, 2000.
- Shim, K.-C., Fontane, D. G., et Labadie, J. W. Spatial decision support system for integrated river basin flood control. *Journal of Water Resources Planning and Management*, 128(3) :190–201, 2002.
- Sreenivasan, K. et Vedula, S. Reservoir operation for hydropower optimization : A chance-constrained approach. *Sadhana*, 21(4) :503–510, 1996.
- Strycharczyk, J. B. et Stedinger, J. R. Evaluation of a “reliability programming” reservoir model. *Water Resources Research*, 23(2) :225–229, 1987.
- Tejada-Guibert, J. A., Johnson, S. A., et Stedinger, J. R. The value of hydrologic information in stochastic dynamic programming models of a multireservoir system. *Water Resources Research*, 31(10) :2571–2579, 1995.
- Tilmant, A., Pinte, D., et Goor, Q. Assessing marginal water values in multipurpose multireservoir systems via stochastic programming. *Water Resources Research*, 44(12) :W12431, 2008.
- Tu, M.-Y., Hsu, N.-S., et Yeh, W. W.-G. Optimization of reservoir management and operation with hedging rules. *Journal of Water Resources Planning and Management*, 129(2) :86–97, 2003.
- Turgeon, A. Stochastic optimization of multireservoir operation : The optimal reservoir trajectory approach. *Water Resources Research*, 43(5), 2007.

- Van Slyke, R. et Wets, R. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, p. 638–663, 1969.
- Vandal, T. Hydroelectricity : green and renewable. *Policy Options*, 2012.
- Watson, J., Woodruff, D., et Strip, D. Progressive hedging innovations for a class of stochastic resource allocation problems. Working paper, Sandia National Laboratories, Albuquerque, 2008.
- Yakowitz, S. Dynamic programming applications in water resources. *Water Resources Research*, 18(4) :673–696, 1982.
- Yi, J., Labadie, J. W., et Stitt, S. Dynamic optimal unit commitment and loading in hydropower systems. *Journal of Water Resources Planning and Management*, 129(5) :388–398, 2003.
- Young, G. K. Finding reservoir operating rules. *Proceedings of the American Society of Civil Engineers*, p. 297–321. 1967.
- Yurtal, R., Seekin, G., et Ardicioglu, G. Hydropower optimization for the lower Seyhan system in Turkey using dynamic programming. *Water International*, 30(4) :522–529, 2005.
- Zeng, Y., Wu, X., Cheng, C., et Wang, Y. Chance-Constrained Optimal Hedging Rules for Cascaded Hydropower Reservoirs. *Journal of Water Resources Planning and Management*, 140(7) :04014010–1–04014010–9, DOI : 10.1061/ (ASCE)WR.1943–5452.0000427, 2014.