

UNIVERSITÉ LOUIS PASTEUR
U.F.R. DE SCIENCES PHYSIQUES
ECOLE DOCTORALE SCIENCES POUR L'INGÉNIEUR

Gestion optimale des réseaux d'eau potable

Florent GUHL

Thèse

en vue de l'obtention du titre de
Docteur de l'Université Louis Pasteur - Discipline : Mécanique

Présentée et soutenue publiquement le 5 février 1999

Directeur de Thèse: Bernard BREMOND

JURY

M. THEPOT Jacques, Président
M.MIRAMOND Marcel, Rapporteur
M.MOREL Paul, Rapporteur
M.GILBERT Denis, Examineur
M.GAGNON Jean-Louis, Examineur

Remerciements

Mes remerciements s'adressent en premier lieu à Denis GILBERT, enseignant-chercheur à l'Ecole Nationale du Génie de l'Eau et de l'Environnement de Strasbourg et Bernard BREMOND, chef de l'unité Ouvrages et Réseaux Hydrauliques au Cemagref de Bordeaux pour leur disponibilité, leurs idées et leur enthousiasme tout au long du déroulement de cette thèse. Que toutes les personnes avec qui j'ai pu travailler à l'ENGEES et au Cemagref trouvent ici l'expression de ma gratitude. Je voudrais en particulier citer Vincent FERSTLER, Yves LE GAT, Philippe ARNOUX ainsi que l'ensemble du service de la documentation de l'ENGEES pour leur aide précieuse lors de la réalisation des recherches bibliographiques.

Cette thèse s'est en grande partie construite autour d'une collaboration entre l'ENGEES, le Cemagref, la Compagnie Générale des Eaux, Groupe Vivendi et Anjou Recherche. Je remercie sincèrement les différentes personnes de la CGE et d'Anjou Recherche qui ont participé à la mise en place et au suivi de cette collaboration. En particulier je suis reconnaissant à Monsieur Dominique OLIVIER d'avoir voulu et su initier cette collaboration afin qu'elle se passe dans les meilleures conditions. Je voudrais aussi remercier pour leur participation à la réflexion et à la mise en oeuvre des travaux de cette thèse Messieurs Jean-Louis GAGNON, Bertrand GOSSOT, Eric LAHAYE, Georges STAWARZ, Frank TEXIER, Philippe DROUET, Thierry PLAISANT et Hervé WACHEUX.

Plusieurs thèmes abordés dans cette thèse nécessitaient des compétences en recherche opérationnelle, nous avons trouvé auprès de quatre universités différentes des interlocuteurs patients et intéressés par les modèles que nous construisions. Je tiens à remercier Messieurs Paul MOREL de l'Université de Bordeaux, Bernard GRANJEAN de l'Université Laval à Québec, Gerd FINKE de l'Université de Grenoble et Jacques THEPOT de l'Université de Strasbourg pour les échanges que nous avons eus et qui ont conduit à des remises en cause méthodologiques conséquentes.

Résumé

Pour que des outils de gestion optimale des réseaux d'eau potable soient utilisés par les gestionnaires de ces réseaux, il est indispensable qu'ils soient les plus complets et les plus robustes possible. Le travail présenté ici a pour but de compléter les modèles d'optimisation élaborés depuis une quinzaine d'années. Il est dicté par une volonté de construire des modèles demandant des temps de calcul courts afin de pouvoir les utiliser dans une gestion en temps réel. C'est pour cette raison que nous privilégions les modèles à base de graphes et de programmation linéaire.

Il s'agit tout d'abord d'améliorer la qualité des entrées des modèles d'optimisation en proposant des modèles de prévision de la demande de bonne qualité mais aussi en intégrant les erreurs résiduelles aux modèles d'optimisation. Le deuxième grand volet de ce travail consiste à prendre en compte dans des modèles linéaires des aspects hydrauliques délicats. Le premier aspect est la nécessaire continuité de fonctionnement de certains ouvrages sur un réseau d'eau potable et le deuxième est l'influence de l'état du réseau sur les coûts et les capacités de transit entre les ouvrages.

La prise en compte des contraintes de continuité est résolue par une méthode de programmation linéaire en nombres mixtes. Les temps de calcul peuvent dans certains cas rester compatibles avec l'objectif d'une gestion en temps réel.

L'influence de l'état du réseau est traité par une méthode de programmation linéaire successive. La convergence de la méthode est mise en défaut sur certains exemples. Sur notre site d'étude la convergence est avérée. Elle s'explique par une dépendance limitée vis-à-vis des états se traduisant par une non-convexité faible.

Mots-clé : Optimisation - Gestion en temps réel - Programmation linéaire - Graphes - Programmation en nombres mixtes - Programmation linéaire successive - Programmation stochastique - Réseaux de neurones - Séries temporelles

Abstract

So that optimal management tools of drinking water networks are used by the managers of these networks, it is essential for them to be as complete and robust as possible. The purpose of work presented here is to supplement the models of optimization worked out since about fifteen years. The main constraint of this work is to build models requiring of the short calculating times in order to be able to use them in management in real time. For this reason the models using graph theory and linear programming is privileged.

First of all, we must improve quality of the inputs of optimization models by offering forecasting models but also by integrating the residual errors into the optimization models. The second large aspect of this work consists in taking into account in linear models two new hydraulic aspects. Firstly, the necessary continuity of operation of certain organs and secondly the influence of the levels in tanks on the costs and the capacities of transit between tanks.

The taking into account of the constraints of continuity is solved by a mixed-integer programming method. The calculating times can in certain cases remained compatible with the objective of a management in real time. The influence of the state of the network is treated by a method of successive linear programming. The convergence of this method is put at fault on certain examples. On our site of study, convergence is showed. This is explained by a weak dependence with respect to the states resulting in a low nonconvexity.

Key-words : Optimization - Real Time Management - Linear Programming - Graph Theory - Mixed-Integer Programming - Successive Linear Programming - Stochastic Programming - Neural Networks - Time Series

Table des matières

Introduction	1
1 Synthèse bibliographique	4
1.1 Les modèles de prévision de la consommation	4
1.1.1 Objectifs de ces modèles	4
1.1.1.1 Minimisation des coûts de production et de distribution	4
1.1.1.2 Mise en évidence de dysfonctionnement du réseau	6
1.1.1.3 Préservation de la ressource naturelle	7
1.1.2 Les méthodes utilisées pour la prévision	7
1.1.2.1 Le pré-traitement de données	8
1.1.2.2 Les facteurs explicatifs de la demande	9
1.1.2.3 Les modèles déterministes	9
1.1.2.4 L'utilisation de méthodes statistiques de régression . . .	11
1.1.2.5 Les séries temporelles	15
1.1.2.6 L'utilisation des méthodes connexionnistes	16
1.1.3 Les réseaux de neurones	17
1.1.3.1 La forme générale d'un réseau de neurones	17
1.1.3.2 Les fondements théoriques	17
1.1.3.3 Le nombre de couches et de neurones	19
1.1.3.4 Les méthodes d'apprentissage	26
1.1.3.5 La qualité de la prévision	27
1.2 La gestion en temps réel des réseaux d'eau potable	31
1.2.1 Objectifs de la gestion d'un réseau d'eau potable	32
1.2.1.1 Minimisation des coûts de production	32
1.2.1.2 Minimisation des coûts d'investissement	33
1.2.1.3 Recherche de fuites	33
1.2.2 Les méthodes d'optimisation utilisées	34
1.2.2.1 Les systèmes expert	34
1.2.2.2 La programmation dynamique discrète	36
1.2.2.3 La programmation non linéaire	38
1.2.2.4 La programmation linéaire	42
1.2.2.5 La programmation linéaire par les graphes	48
1.2.2.6 Les méthodes probabilistes	49

1.2.2.7	Les algorithmes génétiques	52
1.2.2.8	La décomposition - coordination	53
1.2.2.9	Les méthodes neuronales	55
1.3	Conclusion	55
2	Modèles d'optimisation d'un réseau d'eau potable	57
2.1	Modélisation par les graphes	57
2.1.1	Définition générale	57
2.1.2	Construction du graphe	57
2.1.3	Résolution du modèle	59
2.2	Modélisation par la programmation linéaire	60
2.2.1	Horizon et pas de temps	60
2.2.2	Vecteurs de commande	60
2.2.2.1	Définition	60
2.2.2.2	Réduction de l'espace des vecteurs de commande	62
2.2.3	Débits entrant dans les réservoirs	62
2.2.4	Les consommations	63
2.2.5	Les contraintes de volume	63
2.2.6	La fonction économique	63
2.2.7	Ecriture complète du modèle	64
2.3	Conclusion	64
3	Prévision de la demande	66
3.1	Les données	66
3.2	Prévisions à l'heure suivante	67
3.2.1	Construction du modèle de réseau de neurones	67
3.2.2	Résultats du modèle	69
3.2.3	Retour sur le nombre de poids	72
3.3	Construction du modèle de séries temporelles	73
3.3.1	Données de la zone "Vallée"	73
3.3.2	Comparaison des deux méthodes sur les données réelles	74
3.3.3	Comparaison sur des données fictives	75
3.4	Prévisions à plus long terme	76
3.4.1	Les deux modèles de réseaux de neurones	76
3.4.1.1	Modèle à une sortie	76
3.4.1.2	Modèle à 12 sorties	77
3.4.2	Les résultats	77
3.4.2.1	Modèle à une sortie	77
3.4.2.2	Modèle à 12 sorties	78
3.5	Conclusion	78

4	Intégration du caractère stochastique de la demande	82
4.1	Position du problème	82
4.2	Classifications en programmation stochastique	83
4.2.1	Classification d'après la notion de programme	83
4.2.1.1	Formulation serrée	83
4.2.1.2	Formulation en deux étapes	83
4.2.1.3	Problèmes à contraintes probabilistes	83
4.2.1.4	Programme sous forme de distribution	83
4.2.2	Classification d'après le facteur de qualité de la résolution du problème	84
4.2.3	Classification suivant l'aspect dynamique de l'incertitude	84
4.3	Détail des méthodes	85
4.3.1	Formulation serrée	85
4.3.1.1	Méthode par essai	85
4.3.1.2	Méthode par intersection des espaces	85
4.3.2	Problèmes à contraintes probabilistes	86
4.3.2.1	Méthode 1 : utilisation de la distribution de probabilité	86
4.3.2.2	Méthode 2 : programme déterministe convexe équivalent	86
4.3.2.3	Cas des V-Modèle et P-Modèle	87
4.3.2.4	Traitement d'un exemple	87
4.3.3	Problèmes à deux étapes	88
4.3.4	Résolution d'un problème à deux étapes par un programme linéaire déterministe	88
4.3.4.1	Problème de transport	89
4.3.4.2	Programme stochastique	89
4.3.4.3	Programme déterministe équivalent	91
4.3.4.4	Démonstration de l'équivalence	91
4.3.4.5	Interprétation de la méthode	92
4.4	Utilisation des différentes méthodes et application à un réseau exemple	92
4.4.1	L'aspect stochastique de la demande en eau potable	93
4.4.1.1	Les modèles de prévision	93
4.4.1.2	Distribution de la demande	93
4.4.1.3	Le réseau exemple	94
4.4.2	Formulation serrée	94
4.4.2.1	Ecriture du programme	95
4.4.2.2	Contraintes incompatibles	95
4.4.2.3	Application	96
4.4.3	Formulation de contraintes probabilistes	96
4.4.3.1	Présentation probabilisée	97
4.4.3.2	Transformation en programme déterministe	97
4.4.3.3	Application	98
4.4.4	Formulation en problèmes à deux étapes	99
4.4.4.1	Ecriture du programme stochastique	99

4.4.4.2	Programme déterministe équivalent	100
4.4.4.3	Application	101
4.5	Conclusion	103
5	Prise en compte de contraintes de fonctionnement	105
5.1	Position du problème	105
5.2	Utilisation des graphes	105
5.2.1	Représentation de la contrainte de continuité	106
5.2.2	Méthodes de résolution	107
5.2.3	Résultats	107
5.3	Utilisation de la programmation linéaire en nombres entiers	109
5.3.1	Modèle sans respect de la contrainte de continuité	109
5.3.2	Continuité de fonctionnement	109
5.3.2.1	Méthode d'implication	109
5.3.2.2	Implications de continuité	110
5.3.2.3	Variables intermédiaires	112
5.3.2.4	Utilisation des variables intermédiaires	113
5.3.2.5	Variables intermédiaires dans la fonction objectif	114
5.3.3	Exemple d'utilisation de la méthode	115
5.3.3.1	Le réseau exemple	115
5.3.3.2	Les organes de commande	115
5.3.3.3	Données simulées	116
5.3.3.4	Ecriture du modèle	116
5.3.3.5	Contrainte de continuité	116
5.3.3.6	Taille du problème	118
5.3.4	Résolution pratique	119
5.3.4.1	Conditions de résolution	119
5.3.4.2	Utilisation de fonctions coût artificielles	121
5.3.4.3	Utilisation d'une fonction coût réelle	124
5.3.4.4	Choix des paramètres de l'algorithme	126
5.4	Conclusion	128
6	Choix entre plusieurs solutions de même coût	130
6.1	Le problème général	130
6.2	Méthodes de résolution	131
6.2.1	Programmation multi-objectif	131
6.2.1.1	Choix de la fonction objectif	131
6.2.1.2	Ecriture complète du programme	132
6.2.1.3	Résolution	132
6.2.2	Programmation par les graphes	132
6.2.2.1	Structure des coûts	132
6.2.2.2	Méthode proposée	134
6.2.3	Programmation linéaire	135

6.2.3.1	Principe de la méthode	135
6.2.3.2	Limitation du nombre de mises en route	135
6.2.3.3	Optimisation des niveaux	136
6.2.3.4	Ecriture complète du modèle	138
6.3	Conclusion	139
7	Influence de l'état du réseau sur les coûts et les capacités de transit	140
7.1	Ecriture du problème	140
7.2	Formulation par programmation linéaire en nombres entiers	141
7.2.1	Modification des débits entrant	141
7.2.2	Illustration de la dépendance	141
7.2.3	Méthode proposée	142
7.2.3.1	Discrétisation des volumes	142
7.2.3.2	Ecriture complète du programme	144
7.3	Résolution par la programmation linéaire successive	144
7.3.1	Principe de la méthode	144
7.3.2	Ecriture non-linéaire du problème d'optimisation	145
7.3.3	Convergence générale du problème	145
7.3.4	Linéarisation au premier ordre	148
7.3.4.1	Principe de l'algorithme	148
7.3.4.2	Mise en place sur un réseau exemple	149
7.3.5	Utilisation des solutions équivalentes	151
7.3.5.1	Causes de la non-convergence	151
7.3.5.2	Méthode proposée	151
7.3.5.3	Résultats	152
7.3.6	Limite de la convergence	152
7.3.6.1	Explication de la convergence sur le réseau exemple . . .	152
7.3.6.2	Mise en évidence de la non-convergence	153
7.3.6.3	Configuration hydraulique permettant la convergence . .	155
7.4	Conclusion	156
	Conclusion	157

Table des figures

1.1	Réseau de neurones proposé par Mangeas	24
1.2	Exemple de résolution par la programmation dynamique	37
2.1	Exemple de décomposition d'un réservoir en fonction du temps	58
2.2	Réseau exemple pour le traitement de la contrainte de continuité	59
2.3	Graphe du réseau exemple	60
2.4	Séparation des commandes par groupes	62
2.5	Matrice des contraintes du problème linéaire	65
3.1	Prévision à l'heure suivante par réseau de neurones	68
3.2	Distribution des erreurs	71
3.3	Spectre des consommations horaires	72
3.4	Spectre des prévisions	73
3.5	Répartition des erreurs dans la prévision des données artificielles	75
3.6	Prévision à long terme. Réseau de neurones à une sortie	77
3.7	Prévision à long terme. Réseau de neurones à plusieurs sorties	78
3.8	Coefficients de corrélation du modèle à une sortie	79
3.9	Moyenne des erreurs du modèle à une sortie	80
3.10	Coefficients de corrélation du modèle à 12 sorties	80
3.11	Moyenne des erreurs du modèle à 12 sorties	81
4.1	Discrétisation de la fonction de demande	90
4.2	Réseau AEP exemple pour la programmation stochastique	93
4.3	Propriété d'une distribution gaussienne	94
4.4	Exemple de consommations dans une formulation en deux étapes	102
4.5	Trajectoires obtenues en faisant varier le risque	103
5.1	Graphe d'une solution respectant la contrainte de continuité	106
5.2	Fonctionnement de l'usine sans contraintes de continuité	109
5.3	Fonctionnement de l'usine respectant les contraintes de continuité	110
5.4	Illustration de la contrainte de continuité sur un exemple	115
5.5	Arborescence d'un problème en nombres entiers à 3 variables binaires (Minoux, 1983b)	120
5.6	Forme des fonctions coût de validation de la méthode	121
5.7	Résultat avec l'utilisation de la fonction coût $\pi_{2a}^1(t)$	122

5.8	Résultat avec l'utilisation de la fonction coût $\pi_{2b}^1(t)$	122
5.9	Forme de la fonction coût utilisant les tarifs EDF	125
5.10	Résultat avec l'utilisation des coûts réels	125
6.1	Graphe avec mise en évidence des plages de tarifs électriques	133
6.2	Sous-graphes par plages de coût	134
7.1	Réseau AEP illustrant l'influence des niveaux sur les transits	142
7.2	Influence des niveaux sur les débits	143
7.3	Exemple de linéarisation successive présenté dans (Minoux, 1983a)	146
7.4	Structure de l'algorithme de Programmation Linéaire Successive proposé	149
7.5	Sous-réseau Ouest du SIER de Sarralbe	151
7.6	Evolution du critère de convergence pour la linéarisation au premier ordre	152
7.7	Volume dans R5 au cours des itérations	153
7.8	Volume dans R2 entre 19 h et 3 h	154
7.9	Volume dans R5 entre 19 h et 3 h	155

Liste des tableaux

1.1	Répartition de la consommation hebdomadaire selon Grombach	6
1.2	Modèle de demande des CAD selon Peng	11
1.3	Choix des entrées dans le réseau de Canu	19
1.4	Choix des entrées et des sorties du réseau de Wu	20
1.5	Réseaux de neurones de Jain	21
1.6	Comparaison des valeurs de R entre le modèle par réseaux de neurones et le modèle Autorégressif	28
1.7	Comparaison de modèles ARMA et réseaux de neurones	30
1.8	Critères d'évaluation proposés par Jain	30
2.1	Caractéristiques des arcs pour le graphe exemple	58
3.1	Comparaison du nombre de neurones dans la couche cachée	70
3.2	Résultats des prévisions sur les données artificielles	76
4.1	Classification des programmes en programmation stochastique	84
4.2	Résultats de l'optimisation probabilisée	98
4.3	Valeurs de la demande discrète	101
4.4	Influence du paramètre pénalité sur le coût et la consommation	102
5.1	Comparaison des trois méthodes de résolution de la contrainte de continuité par les graphes	108
5.2	Nombre de contraintes dans le modèle sans contraintes de continuité	119
5.3	Comparaison du nombre de variables dans chaque modèle	123
5.4	Nombre de variables et de contraintes après presolve	123
5.5	Calculs supplémentaires dus à la contrainte de continuité	124
5.6	Temps de calcul avec les coûts reels	126
7.1	Vecteurs de commande du sous-réseau ouest	150
7.2	Evolution du critère de convergence de la deuxième méthode	152
7.3	Evolution du critère de convergence après suppression des influences multiples	156

Introduction

Les réseaux d'eau potable sont de plus en plus souvent équipés de systèmes de télégestion des installations et des commandes. L'accroissement de ces équipements accompagne généralement une augmentation de leur complexité. En effet, pour des raisons de sécurité mais aussi de rentabilité, beaucoup de réseaux résultent de l'interconnexion de sous-réseaux qui étaient auparavant gérés de façon autonome. Ceci augmente la difficulté de choisir au cours d'une journée les consignes à donner aux différents organes d'un réseau d'eau potable.

La complexité de cette gestion vient aussi de la prise en compte par les gestionnaires des réseaux de deux objectifs contradictoires qui sont la sécurité d'approvisionnement et la minimisation des coûts de production et de transport de l'eau. Pour aider les opérateurs dans la gestion des réseaux, de nombreux outils d'automatisation ont été conçus depuis une vingtaine d'années. Ils peuvent être regroupés en deux catégories : ceux faisant appel à des systèmes experts s'inspirant initialement des méthodes utilisées par les hydrauliciens et ceux reposant sur des méthodes d'optimisation de la recherche opérationnelle.

Afin de proposer des méthodes les plus transposables possibles, nous nous sommes intéressé aux méthodes d'optimisation reposant sur la modélisation hydraulique des flux associée à des outils de programmation mathématique. L'étude présentée s'inscrit dans la continuité de travaux menés depuis la fin des années 80 en collaboration entre l'Ecole Nationale du Génie de l'Eau et de l'Environnement et le Cemagref. Les premières étapes de ces travaux antérieurs ont permis de montrer la faisabilité et les limites d'utilisation des outils de programmation mathématique. En particulier, un choix s'est imposé en direction de l'utilisation de la programmation linéaire par les graphes. Cette méthode permettait d'envisager des temps de calcul suffisamment courts pour être compatibles avec une gestion en temps réelle qui constitue un objectif important. Par ailleurs, la programmation linéaire par les graphes semblait permettre une modélisation fidèle de la plupart des configurations hydrauliques rencontrées dans la réalité. Au fur et à mesure

de l'avancement des travaux il est apparu que l'importance de certains phénomènes physiques avait été sous-estimée. De plus, avec les années, les gestionnaires de réseaux ont accru leurs exigences quant à la finesse des modélisations à proposer.

Cette étude vise donc à proposer un modèle d'optimisation du fonctionnement d'un réseau d'eau potable sur 24 heures permettant de tenir compte des deux évolutions constatées. Dans ce modèle nous cherchons à améliorer la prévision de la demande horaire des consommateurs ainsi que l'intégration la plus complète possible de cette demande dans les modèles d'optimisation. Nous cherchons aussi à intégrer des phénomènes non pris en compte précédemment comme les contraintes de continuité et l'influence de l'état du réseau sur les transits hydrauliques.

Nous avons également voulu maintenir l'objectif de gestion en temps réel des réseaux. C'est pour cette raison que la plupart des méthodes que nous présentons ici repose sur la programmation linéaire qui permet d'envisager des temps de calcul assez faibles sans pour autant demander une simplification trop forte.

Nous débutons la présentation de notre étude par un bilan des travaux menés jusqu'à maintenant dans le domaine de la prévision de la demande ainsi que dans celui de l'optimisation de grands systèmes de transport et en particulier des réseaux d'eau potable.

Nous montrons alors la façon dont nous construisons les modèles d'optimisation d'un réseau d'eau potable. Nous présentons en fait une méthode utilisant la programmation par les graphes et une méthode utilisant la programmation linéaire.

Toujours en nous appuyant sur le tour d'horizon de la première partie, nous comparons la pertinence des modèles de séries temporelles et par réseaux de neurones pour bâtir notre propre modèle de prévision de la demande à partir des données dont nous disposons.

Même si ce modèle améliore sensiblement la qualité de la demande utilisée lors de la phase d'optimisation, il est apparu judicieux de tenir compte de l'incertitude de prévision pour rendre les modèles d'optimisation plus complets. Les méthodes de prise en compte de l'aspect stochastique de la demande permettent en particulier de quantifier le niveau de sécurité choisi par le gestionnaire du réseau.

Nous verrons alors comment gérer dans nos modèles d'optimisation des contraintes non prises en compte jusqu'à présent dans les modèles d'optimisation appelées contraintes de continuité. Elles sont liées à l'obligation sur certains ouvrages d'avoir un fonctionnement par plages continues. Ce chapitre montre en particulier que la programmation linéaire en nombres mixtes peut apporter une réponse à ce problème.

La structure des coûts unitaires de production et de transport de l'eau étant liée aux tarifs électriques, nous rencontrons régulièrement des situations où il est possible de modifier les commandes à passer aux organes du réseau au cours d'une même plage de tarif sans modifier le coût total de la solution. Nous proposons donc des améliorations de nos modèles de façon à permettre un choix parmi ces solutions.

Enfin, nous présentons des méthodes permettant de tenir compte de l'influence de l'état du réseau sur les transits entre ouvrages sans utiliser la programmation dynamique ou non-linéaire. En particulier, nous montrons les possibilités et les limites de l'utilisation de la programmation linéaire successive.

Chapitre 1

Synthèse bibliographique

1.1 Les modèles de prévision de la consommation

La plupart des références examinées concernent la prévision de consommation en eau potable. Toutefois, des auteurs traitant d'autres domaines techniques (électricité ou réseaux de chauffage urbain) fournissent également des méthodes intéressantes.

1.1.1 Objectifs de ces modèles

Les méthodes présentées ici diffèrent suivant l'objectif initial. Il peut notamment exister des différences significatives sur l'horizon de prévision qui est considéré.

On dégage trois objectifs principaux dictant la réalisation d'outils de prévision de la consommation.

1.1.1.1 Minimisation des coûts de production et de distribution

Comme nous le verrons dans le chapitre 2 consacré à l'optimisation des réseaux d'AEP, il est essentiel, pour mettre en oeuvre un modèle d'optimisation des coûts, de disposer de bons paramètres en entrée. Cette entrée est constituée dans le cas des réseaux d'eau potable d'une estimation de la demande sur la durée de l'horizon d'optimisation.

Néanmoins, certains auteurs ont choisi de simplifier l'aspect prévision de la consommation dans leur étude sur la minimisation des coûts proportionnels.

Cela est le cas dans (Carpentier, 1993) où les auteurs utilisent des courbes de consommation types. Ils signalent que les volumes journaliers prédits sont susceptibles d'être entachés de près de 20 % d'erreur. Toutefois, ils se prémunissent contre les conséquences de ces erreurs en envisageant de nouveaux calculs d'optimisation en cours d'horizon, en

fonction d'un écart éventuel entre les valeurs attendues et les valeurs constatées. Ces calculs peuvent avoir lieu toutes les heures.

C'est également cet a priori qui est retenu dans (Carpentier, 1983), (Gilbert, 1988) et (Gilbert, 1990).

Souvent ce type de démarche est accompagné d'une étude de la sensibilité de l'erreur commise sur la prévision. La mesure de la sensibilité consiste à apprécier la perte d'optimalité de la solution dans le cas où les consommations réellement enregistrées divergent de celles prévues.

Pour illustrer cette mesure de sensibilité, nous pouvons détailler la méthode utilisée dans (Amoretti, 1996) où la fonction définissant la consommation horaire prend la forme de l'expression (1.1).

$$Q(h) = Q_m.K(h).a \quad (1.1)$$

avec

- Q_m , débit moyen journalier (m^3/h)
- $K(h)$, coefficient de modulation horaire
- a , variable aléatoire.

La variable aléatoire " a " permet de simuler l'erreur de prévision sur la consommation. On suppose que cette erreur peut atteindre 30 % sur les réseaux de moyenne importance et 50 % sur les petits réseaux. On réalise donc plusieurs centaines de simulations où le coefficient " a " prend des valeurs aléatoires comprises entre 0.7 et 1.3 ou entre 0.5 et 1.5 suivant la taille du réseau. C'est à partir d'un traitement statistique de ces simulations que l'on déduit la perte d'optimalité.

D'autres auteurs n'abordent le problème que de façon partielle en considérant une partie de la fonction de demande comme un paramètre établi une fois pour toutes. C'est le cas de (Gravel, 1996) où l'on considère un spectre de la répartition de la consommation sur une journée valable quelle que soit la journée. L'objectif du travail décrit consiste à anticiper d'éventuels problèmes sur le réseau. Pour ce faire, il utilise des équations bilans pour chaque réservoir. C'est dans ces équations bilans que l'auteur a besoin de coefficient pondérant les sorties en fonction de l'heure de la journée.

C'est également ce choix qui est fait dans (Grombach, 1982) où la méthode employée consiste à calculer une fois pour toutes les pourcentages horaires pour chaque zone de consommation. A partir de là, la prévision de volume se fait en tenant compte du volume déjà consommé dans les heures précédentes. Aucune distinction n'est faite entre les jours

fériés et ouvrables et entre les jours secs et humides car la distinction doit se faire directement via le volume déjà consommé.

Pour la prévision hebdomadaire l'auteur répartit la demande selon les chiffres de la table 1.1.

Jour de la semaine	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
Part de la consommation hebdomadaire	17 %	14 %	14 %	14 %	20 %	12 %	9 %

Tableau 1.1 : Répartition de la consommation hebdomadaire selon Grombach

De la même façon, dans (Ormsbee, 1994) seule la prévision de la demande globale sur 24 heures est effectuée (voir paragraphe 1.1.2.4). La répartition sur les 24 tranches horaires de la journée se fait par l'utilisation de spectres dépendant du jour de la semaine et de la saison de l'année.

1.1.1.2 Mise en évidence de dysfonctionnement du réseau

Sur des réseaux de très grande taille, comme ceux de la région parisienne, les outils de prévision de la consommation sont souvent utilisés pour donner à tout instant l'image attendue du réseau en terme de débit et de pression. Pour cela on utilise un modèle hydraulique qui, couplé à cette prévision de consommation, permet de comparer les mesures à la situation théorique attendue. Ainsi dans (Gauriau, 1996) les auteurs décrivent les outils de prévision mis en place par le Centre des Mouvements de l'Eau sur le réseau du SEDIF (Syndicat des Eaux d'Ile de France). Un calcul est effectué toutes les 2,5 mn. Ce calcul est en fait un bilan d'entrée/sortie sur chaque sous-réseau. Il fournit comme résultat la somme des consommations et des fuites par sous-réseau.

A partir de ces consommations constatées, un algorithme prédit les volumes journaliers et leur décomposition horaire. Le volume journalier est prédit en faisant une moyenne glissante des volumes des 7 derniers jours affectée d'un coefficient dépendant le jour de la semaine.

Grâce aux prévisions réalisées, il est possible de mettre en place des seuils dynamiques de tolérance pour détecter des anomalies. Une méthodologie de surveillance identique est également reprise dans (Crommelynck, 1991).

Les phénomènes qui sont détectés par ces outils peuvent aussi bien être une casse qu'une surconsommation. Il est utile de pouvoir les discriminer. C'est dans ce cadre

qu'a eu lieu l'étude décrite dans (Bissery, 1994) pour détecter les fuites sur le réseau de la Courly (Communauté Urbaine de Lyon). Dans cette étude, la modulation des consommations sur la journée est réalisée à l'aide d'un spectre de consommation. Ce spectre est construit sans tenir compte des gros consommateurs qui sont traités à part. De plus, ce spectre est recalé tous les ans pour tenir compte des évolutions de comportement des consommateurs.

1.1.1.3 Préservation de la ressource naturelle

Des études menées au cours des dernières années dans l'Ouest américain sont dictées par une volonté des autorités locales de diminuer l'effet de pointe de consommation. L'objectif est alors double : baisser le débit instantané maximal demandé par un réseau et baisser aussi le volume à fournir sur une période donnée (le jour, la semaine, le mois, la saison ou l'année). Le détail des moyens conséquents mis en oeuvre pour réaliser ce type d'étude est donné dans (Rothstein, 1993), (Rothstein, 1991) et (Rhoades, 1995) sur le cas particulier de la ville d'Austin, Texas.

Les méthodes utilisées seront détaillées par la suite. Elles peuvent déboucher sur des outils logiciels comme IWR-MAIN dont la version 6.0 est présentée dans (Dziegielewski, 1993a). Ce logiciel permet de tester des mesures qui pourraient être prises pour diminuer les pics de consommation sur un secteur. Les mesures envisagées sont de deux natures radicalement différentes. Il s'agit soit de décrets imposant un limiteur de débit chez les abonnés soit d'une politique tarifaire incitant à consommer à certains moments plutôt qu'à d'autres. Cette deuxième méthode semble proche de ce que nous connaissons en France avec les tarifs électriques.

1.1.2 Les méthodes utilisées pour la prévision

Nous avons orienté nos recherches vers le choix des paramètres d'entrée utilisés par les différents auteurs dans la construction de leur modèle. C'est, en effet, un domaine où règne une grande diversité et dans lequel nous aurons à faire un choix par la suite.

En parallèle nous nous sommes intéressés aux modèles réalisant une prévision (de consommation en AEP ou d'une autre fonction de demande) à l'aide d'outils connexionnistes. Avant cela, nous nous intéresserons aussi aux méthodes déterministes ou relevant de statistiques classiques.

1.1.2.1 Le pré-traitement de données

Avant de voir les modèles utilisables pour la prévision de la demande, il est intéressant de faire le point sur les méthodes de pré-traitement qui permettent d'assurer une meilleure qualité des données en entrée de ces modèles.

Dans (Abdollahzadeh, 1997), l'auteur présente des méthodes de validation de données par équilibrage de bilans. Il propose une méthode matricielle permettant de dégager quatre types de variables : les variables redondantes, mesurées mais non estimables, non mesurées mais déductibles et, enfin, non mesurées et non déductibles.

Pour détecter des valeurs incohérentes, il met en place deux types de tests :

1. l'analyse des écarts de fermeture des équations de bilan (test a priori) permet de détecter s'il existe globalement des erreurs de mesures puis, le cas échéant et avec l'aide d'un autre test, d'incriminer tel ou tel point de mesure en utilisant notamment les redondances mises en évidence précédemment.
2. l'analyse des termes correctifs normalisés (test a posteriori) permet de calculer des erreurs estimées et vérifie que les termes correctifs normalisés sont dans un intervalle de confiance déduit d'un seuil de confiance choisi arbitrairement.

L'application proposée dans cette thèse porte sur le réseau d'eau potable de la Communauté Urbaine du Grand Nancy.

On pourra également se reporter à (Rakotondrabe, 1988) et (Ragot, 1993) où les auteurs abordent clairement le choix optimal de données manquantes (positionnement de capteurs).

Dans (Cohen, 1991) l'étude porte sur l'observabilité d'un réseau d'eau potable obtenue en ne mesurant qu'un nombre restreint de paramètres. Les auteurs proposent de mettre en oeuvre des méthodes basées sur la théorie des graphes (les graphes augmentés) inspirées des travaux de T. Ozawa.

Un des outils permettant de consolider des données incomplètes ou dont la qualité est incertaine est la réconciliation de données.

Dans (Du, 1997), les auteurs abordent la réconciliation de données à l'aide de réseaux de neurones. Ils présentent, tout d'abord, un réseau avec apprentissage supervisé. La méthode nécessite de connaître la sortie pour chaque entrée. Ils contournent cette difficulté par d'autres méthodes de réconciliation de données qu'ils utilisent dans un calcul hors ligne.

Ils proposent, par la suite, une autre méthode qui consiste en un modèle auto-associatif de (Kramer, 1992) ou de (Dong and McAvoy, 1995). Le modèle présenté est

constitué de cinq couches (donc trois couches cachées) comportant une couche centrale beaucoup plus petite que les autres (pouvant ne contenir qu'un seul neurone). La fonction objectif qui est à minimiser comporte deux termes : le premier permet de minimiser l'erreur entre les entrées et les sorties calculées et le deuxième terme minimise l'écart sur chaque équation contrainte du système. La difficulté réside alors dans le choix des coefficients de pondération des deux termes. Un essai réel a montré qu'au-delà d'une certaine valeur du coefficient devant les équations bilans il n'y a plus aucune amélioration de ce critère, alors que le critère d'erreur continue de se détériorer. Cette constatation permet d'arrêter un choix optimal d'architecture du réseau.

1.1.2.2 Les facteurs explicatifs de la demande

L'inventaire des variables qui peuvent être utilisées en entrée d'un système de prévision de la demande peut être assez long. Nous avons choisi de les présenter en utilisant la distinction faite dans (Mangeas, 1993) entre les variables endogènes et les variables exogènes.

1.1.2.2.1 Les variables endogènes Il s'agit des variables qui sont de la même nature que la sortie recherchée. Dans le cas de la prévision de consommation, en particulier, il s'agit d'utiliser la consommation mesurée aux pas de temps qui précèdent celui de la prévision. Ainsi, dans (Wu, 1995) les auteurs construisent leur modèle sur ce qu'ils appellent le principe fondamental de la prévision et qu'ils énoncent de la façon suivante : "la valeur d'une quantité à un pas de temps donné est principalement déduite de la valeur au pas de temps précédent".

1.1.2.2.2 Les variables exogènes Elles traduisent l'influence de paramètres extérieurs sur la consommation. Leur choix dépend de la réflexion du concepteur du modèle et du lieu où est réalisée l'étude. La méthode globale qui est employée consiste à établir une liste exhaustive des paramètres qui selon l'auteur peuvent influencer la consommation (températures passées et futures, jour de la semaine, pluviométrie,...) puis à éliminer les paramètres peu influents à la suite d'une étude statistique ou à l'aide d'un réseau de neurones.

1.1.2.3 Les modèles déterministes

Dans ce type d'approche, pas très courante dans notre domaine, il s'agit de connaître précisément l'usage qui est fait de l'eau distribuée chez l'utilisateur. Ainsi dans (Dziegie-

lewski, 1993b) l'auteur applique un principe de base qui est celui d'une décomposition d'un volume global. Cette décomposition se fait à l'aide de la connaissance de l'utilisation réelle de l'eau par les particuliers et les collectivités. Cela nécessite de connaître le niveau d'équipement en appareils consommant de l'eau de chaque foyer. Dans la prévision de la demande, deux niveaux sont distingués : le nombre de consommateurs et la quantité d'eau consommée par consommateur. L'auteur constate que certains facteurs ont une influence en temps réel (la température) alors que d'autres ont une influence plus lente dans le temps (le prix de l'eau). De plus, l'évolution des paramètres n'est pas non plus la même : certains évoluent lentement comme la densité de population sur un secteur alors que d'autres peuvent varier à court terme comme la météo ou le prix de l'eau.

Ainsi une formule est établie qui permet de lier la prévision de demande aux variables suivantes : le salaire moyen du ménage, le nombre moyen de personnes, la densité moyenne de population, la température maximale journalière, la pluie, le prix de l'eau et de l'assainissement et le taux d'impôt.

Pour pouvoir déterminer la consommation d'eau de chaque foyer l'auteur utilise des mesures électriques qui sont d'un accès plus facile. De plus, l'auteur détermine pour chaque appareil consommateur d'eau la quantité moyenne à chaque utilisation (pour les toilettes par exemple).

N'ayant pas trouvé de références suffisamment détaillées utilisant ce type de modélisation de la demande, nous avons choisi de présenter la démarche utilisée dans la prévision de consommation des réseaux de chauffage à distance (les CAD) de la ville de Lausanne. Cet aspect de prévision de la demande constitue une partie de (Peng, 1995).

Le modèle utilisé combine à la fois une démarche déterministe et statistique. L'auteur distingue, en effet, trois types de modèles :

- les modèles analytiques qui consistent en un bilan thermique du bâtiment ;
- Les modèles statistiques qui analysent un grand nombre de données ;
- Les modèles utilisant des indices énergétiques qui combinent les deux approches précédentes (les indices représentatifs permettant de caractériser la consommation d'énergie de chauffage des bâtiments de différentes catégories).

C'est un modèle du troisième type qui est développé dans ce document. En fait, l'estimation de la demande est dans un premier temps réalisée de façon analytique avec un bilan complet des entrées et des sorties dans un appartement. Dans un deuxième temps l'équation est simplifiée pour aboutir à une demande annuelle dépendant du nombre de logements dans l'immeuble et de sa surface. C'est, alors, une régression linéaire multiple

qui permet, à partir d'une démarche statistique (basée sur 266 bâtiments), d'établir la relation (1.2).

$$Q_{th} = a.NL + b.SEV \quad (1.2)$$

avec NL le nombre de logements dans l'immeuble et SEV la surface enveloppe du bâtiment.

Il est à noter que pour affiner ce travail, l'échantillon de départ a été découpé en trois groupes en fonction de deux variables : le nombre d'étages (NE) et la date de construction de l'immeuble (DATE) de façon à obtenir un couple (a,b) pour chacune des classes. La variable DATE correspond à des classes d'âges permettant d'intégrer la qualité de l'isolation. Cette classification rend la méthode plus pertinente puisque l'on obtient les résultats du tableau 1.2 qui montrent bien des modèles différents.

Echantillon	DATE	NE	Valeur de a	Valeur de b
Complet	1-6	1-10	3,67	0,16
n° 1	1-2	3-6	5,06	0,11
n° 2	3-4	2-6	4,57	0,14
n° 3	5-6	1-10	2,50	0,18

Tableau 1.2 : Modèle de demande des CAD selon Peng

En électricité on peut également rencontrer des modèles de prévision de la demande construits sur une base déterministe. Ainsi dans (Papon, 1994), l'auteur conçoit un modèle qui permet de prédire la consommation d'électricité en fonction de la météorologie. Là aussi le modèle est à la fois déterministe et statistique dans la façon d'obtenir les relations liant les paramètres.

L'auteur utilise un algorithme, appelé algorithme de Hook et Jeeve qui revient à la minimisation d'un critère d'écart quadratique, pour obtenir des relations non linéaires. Ces relations peuvent faire intervenir des effets de seuil. Le plus significatif est celui intervenant par rapport à la température où il n'y a pas augmentation de la consommation électrique en dessous d'une certaine valeur de température car ce sont d'autres modes de chauffage qui sont alors utilisés par les consommateurs pour compléter le chauffage électrique.

1.1.2.4 L'utilisation de méthodes statistiques de régression

Dans le paragraphe consacré aux variables exogènes nous avons vu qu'il était souvent nécessaire de faire le tri entre les variables qui paraissent a priori explicatives et celles qui

le sont réellement. Afin d'appréhender le poids des différentes variables, il est possible de réaliser une régression multiple. C'est cette méthode qui est employée dans (Ormsbee, 1994). Les variables exogènes conservées sont la météo de la veille, la prévision météorologique du jour à venir et le type de saison. L'auteur mêle ces variables exogènes aux consommations des jours passées pour aboutir aux entrées de son système prédisant la consommation du jour à venir.

Une des méthodes classiques en statistique consiste à partitionner le domaine en classes homogènes. Dans (Gauriau, 1996), par exemple, on utilise des données de consommation sur plusieurs années pour aboutir à des coefficients caractérisant le jour de la semaine. Ces coefficients sont au nombre de trois pour ce réseau urbain de la région parisienne : celui du samedi, celui du dimanche et celui des jours de semaine. Le volume journalier est alors prédit en faisant une moyenne glissante des volumes des 7 derniers jours affectée de ce coefficient journalier.

Dans (Boniface, 1992), l'auteur utilise une AFC (analyse factorielle des correspondances) pour déterminer les variables exogènes qu'il faut utiliser en entrée d'un modèle de prévision. Ces variables sont uniquement utilisées pour une prévision du volume journalier. Ces variables sont les suivantes :

- consommation de la veille et du même jour dans la semaine précédente,
- nombre de jours consécutifs sans pluie,
- place du jour dans la semaine et dans l'année,
- type de journée (jour férié, vacances scolaires,...),
- température de la veille,
- prévisions de température et de pluviométrie pour le jour à venir.

Les auteurs de (Kim, 1985) choisissent les variables significatives de façon assez semblable. L'article s'intéresse à la prévision de consommation annuelle sur les six quartiers de la ville de Taegu en Corée. La méthode employée est fondée sur des régressions linéaires et logarithmiques.

Dans un premier temps les auteurs recensent différentes variables explicatives de la demande globale et unitaire par quartier. Les variables, déterminées a priori, sont les suivantes :

- population totale du quartier,
- population effectivement raccordée,
- pourcentage de la population desservie,

- superficie desservie,
- pourcentage de la superficie desservie,
- densité de population,
- revenu par personne,
- prix de l'eau,
- nombre de commerces,
- nombre d'employés de l'industrie.

Dans un deuxième temps, une ACP (Analyse en Composantes Principales) permet de dégager trois axes principaux. Enfin, les régressions peuvent être obtenues en utilisant des variables indépendantes.

Dans (Lannuzel, 1990), la prévision de consommation porte sur une partie de la commune de Neuilly-sur-Marnese. Cette prévision se fait en deux parties : le calcul du volume total de la journée puis l'application d'un profil de consommation type. Pour dégager les principaux profils, une ACP a été effectuée. Celle-ci a permis de définir 5 périodes dans l'année :

1. hiver : du 1er décembre au 28 février,
2. printemps : du 1er mars au 31 mai,
3. été : du 1er juin au 31 juillet,
4. août,
5. automne : du 1er septembre au 30 novembre.

Le seul facteur explicatif retenu pour chacune de ces saisons est la température journalière maximale prévue, notée aussi T_p .

La prévision du volume total de la journée se fait alors de la façon suivante :

- on calcule pour chaque jour la consommation réduite C_r . A titre d'exemple la relation (1.3) illustre le calcul du lundi 2 novembre

$$C_r = \frac{\text{consommation brute du lundi 2 novembre}}{\text{moyenne des consommations des lundis compris entre 01/09 et 30/11}} \quad (1.3)$$

- on détermine trois coefficients a , b et c à l'aide du logiciel STATGRAF pour écrire l'équation suivante : facteur-jour(J) = $a * C_r(J - 1) + b * T_p + c$
- Il suffit, ensuite, de multiplier le facteur-jour par la moyenne des consommations du jour pour avoir la prévision globale.

Pour la gestion des profils, les auteurs ont utilisé une AFC. Celle-ci a révélé l'existence d'un premier axe discriminant les dimanches et jours fériés des autres jours et d'un deuxième axe séparant le samedi des jours ouvrables. Les auteurs ont donc commencé la prévision en séparant ces trois domaines avant de se rendre compte que le lundi matin et le vendredi soir étaient mal prédits. Ils ont finalement choisi de définir un profil par jour de la semaine.

Nous avons vu au paragraphe 1.1.1.3 que certains auteurs cherchent à quantifier l'impact de mesures de protection de la ressource. Ainsi dans (Little, 1991), on utilise des moyennes mensuelles pour caractériser l'écart entre la demande actuelle et celle résultant des mesures de protection.

Le modèle de prévision qui est construit ici comprend plusieurs parties :

- une partie linéaire traduisant la tendance générale d'évolution dans le temps : $a_0 + a_1 t$.
- une partie polynômiale (de degré trois) traduisant l'impact des cycles saisonniers (les termes élevés aux différentes puissances et traduisant le temps sont des différences entre t et un terme t_0 origine de référence, le 1er janvier par exemple) : $a_2 (t - t_0) + a_3 (t - t_0)^2 + a_4 (t - t_0)^3$.
- une partie utilisant des indicatrices (0/1) traduisant les cycles hebdomadaires (un paramètre pour chaque jour de la semaine) : $a_5 M_t + a_6 T_t + a_7 W_t + a_8 TH_t + a_9 F_t + a_{10} SA_t$.
- une autre partie linéaire utilisant sept paramètres de température définis chacun comme la différence entre la température maximale du jour et la moyenne des températures maximales du mois et ceci pour le jour j et les six autres jours de $j - 1$ à $j - 6$: $b_0 DT_t + b_1 DT_{t-1} + \dots + b_6 DT_{t-6}$.

Afin de réduire le nombre de paramètres, les auteurs expriment les sept coefficients b_j comme des polynômes de degré deux (modèle de Almon) en fonction de j selon l'équation (1.4). Ceci revient à réduire de sept à trois le nombre de paramètres.

$$b_j = g_0 + g_1 j + g_2 j^2, j \in \langle 1; 6 \rangle \quad (1.4)$$

- La dernière partie du modèle traduit l'influence des précipitations. Elle fait intervenir deux types de variables. Le premier groupe de variables est construit de façon strictement analogue à la partie du modèle traduisant l'influence des températures. Il fait donc intervenir les précipitations du jour et des six jours précédents en utilisant le modèle de Almon. Le deuxième paramètre traduisant l'influence des

précipitations est un terme donnant le nombre de jours consécutifs où la précipitation n'a pas dépassé une valeur seuil.

Les essais réels qui ont été menés sur les données des villes de Camden et Durham aux États-Unis ont permis de préciser certains aspects du modèle :

- il est plus intéressant de construire deux modèles séparés pour l'été et pour l'hiver. Ceci revient à faire un découpage en deux classes qui sont plus homogènes et pour lesquelles on obtient des coefficients différents.
- La variable “jour de la semaine” peut se résumer à une variable qui serait “dimanche/autre jour”. En effet, les coefficients intervenant devant les indicatrices du jour de la semaine sont très faibles pour les jours allant du lundi au samedi.

1.1.2.5 Les séries temporelles

Le type de données que nous utilisons semble propice à l'utilisation de séries temporelles. L'avantage de ces méthodes est de décomposer les données suivant les différents facteurs explicatifs comme les périodicités (saison et périodicité hebdomadaire).

Dans (Gourieroux, 1995), les auteurs consacrent tout un chapitre à la désaisonnalisation dans lequel ils présentent la méthode des moindres carrés. Lorsque les résidus sont autocorrélés, ils appliquent la méthode des moindres carrés généralisés à la place des moindres carrés ordinaires. Pour savoir s'il y a autocorrélation d'ordre 1, les auteurs présentent le test de Durbin-Watson.

Un autre chapitre est consacré aux méthodes de décomposition du type “moyenne mobile”. Les auteurs présentent, en particulier, la moyenne arithmétique. Ils détaillent ensuite diverses moyennes mobiles qui permettent de ne conserver que certaines tendances. On peut ainsi isoler la tendance générale ou la saisonnalité.

Les auteurs présentent, enfin, quelques processus univariés, notamment ARMA (Auto-Regressive Moving Average) et ARIMA (Auto-Regressive Integrated Moving Average). La définition complète d'un modèle ARMA est alors donnée pour un processus stationnaire, c'est-à-dire un processus avec moyenne et covariances indépendantes du temps.

”Un processus stationnaire X admet une représentation ARMA (p, q) minimale s'il satisfait la relation (1.5).”

$$X_t + \varphi_1 X_{t-1} + \dots + \varphi_p X_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} \Leftrightarrow \Phi(L) X_t = \Theta(L) \varepsilon_t \quad (1.5)$$

Dans la relation (1.5), on peut expliciter :

i) $\varphi_p \neq 0, \theta_q \neq 0$,

- ii) les polynômes Φ et Θ ont leurs racines de modules strictement supérieurs à 1,
- iii) Φ et Θ n'ont pas de racines communes,
- iv) $\varepsilon = (\varepsilon_t, t \in \mathbb{Z})$ est un bruit blanc, de variance $V\varepsilon_t = \sigma^2 \neq 0$,
- v) L est l'opérateur retard qui associe au processus $(X_t, t \in \mathbb{Z})$ le processus $(Y_t, t \in \mathbb{Z})$ tel que $Y_t = LX_t = X_{t-1}$

Lorsque l'hypothèse d'une série stationnaire n'est pas vérifiée, on introduit des modèles où la différence est stationnaire (modèles ARIMA). A la suite, les auteurs présentent la méthode de prévision Box et Jenkins qui utilise les processus ARIMA ainsi que le critère BIC permettant de choisir parmi des modèles ARIMA.

D'autres ouvrages présentent le cadre général des séries temporelles comme (Wonnacott, 1984), (Coutrot, 1984) et (David, 1989). Dans ce dernier, les auteurs détaillent les filtres permettant de lisser les données. Ils présentent notamment les filtres de moyenne mobile qui s'écrivent :

$$x_t^* = \sum_{i=-p_1}^{x=p_2} \theta_i x_{t+i} \quad (1.6)$$

Enfin, ils expliquent ce qu'est l'analyse spectrale dans les séries temporelles. Celle-ci permet de mettre en avant des périodicités difficiles à repérer sans cette analyse. En annexe de leur ouvrage, les auteurs présentent l'exemple de la prévision du niveau d'eau dans le St Laurent avec une bonne illustration de l'information que donnent les corrélogrammes.

1.1.2.6 L'utilisation des méthodes connexionnistes

L'utilisation de réseaux de neurones pour réaliser des prévisions a connu, comme les réseaux de neurones eux-mêmes, un regain d'intérêt depuis la fin des années 80. La plupart des utilisateurs de ces techniques a utilisé des techniques de statistique classique par le passé. C'est pourquoi leur présentation des avantages et inconvénients de chaque méthode est particulièrement intéressante. On pourra notamment se référer à (Canu, 1996) où les deux domaines sont mis en parallèle. Dans cette référence, un bref rappel historique permet de balayer les évolutions des méthodes depuis le perceptron de Rosenblatt en 1958 jusqu'aux dernières applications en reconnaissance de formes. Ce tour d'horizon montre que les travaux ont été en grande partie suspendus entre 1969 et 1985, ce qui explique l'impression de présentation d'un domaine nouveau. Le chapitre suivant (1.1.3) sera entièrement consacré à la présentation de la structure de fonctionnement des réseaux de neurones ainsi qu'à la façon dont ils sont utilisés pour effectuer de la prévision.

1.1.3 Les réseaux de neurones

1.1.3.1 La forme générale d'un réseau de neurones

Pour présenter la forme générale prise par un réseau de neurones, nous pouvons reprendre la définition donnée dans (Guillon, 1991) : “Un réseau de neurones artificiels est typiquement constitué d'un très grand nombre de processeurs simples organisés en une architecture massivement parallèle, auto-organisée et auto-adaptable. Chaque processeur, qu'on appelle par analogie une cellule, reçoit les données qui proviennent des cellules situées sur la couche précédente et envoie les résultats de son traitement aux cellules situées sur la couche suivante.” Le mode de fonctionnement d'une telle architecture est également décrit dans ce même article : “Le poids des connexions change chaque fois que le système apprend, c'est-à-dire ajuste sa réponse (la sortie du réseau) à une sortie jugée désirable.”

On retrouve également dans une thèse (Bigot, 1995) des explications détaillées sur la structure des réseaux de neurones artificiels. Dans ce document l'auteur utilise les réseaux de neurones pour prédire à tout moment l'état attendu d'un réseau d'eau potable, celui de la Communauté Urbaine de Lyon. Dans cette thèse, la distinction est faite entre des réseaux de neurones à apprentissage supervisé et à apprentissage non supervisé. Nous nous situerons dans le cadre d'un apprentissage supervisé lorsque nous disposerons d'une série de données comportant des entrées et les sorties correspondantes. L'apprentissage non supervisé permet, à défaut de connaître les sorties attendues, de discriminer ces sorties en plusieurs domaines homogènes et distincts entre eux (à la manière des méthodes de classification en analyse de données).

1.1.3.2 Les fondements théoriques

Dans la littérature, les réseaux neuronaux ne bénéficient pas des mêmes justifications théoriques que les méthodes de statistique classique. Néanmoins certains auteurs donnent des éléments de justification. Ainsi dans (Canu, 1995) la formulation du problème de prévision non linéaire est donnée. De même, dans (Boucheron, 1995), les auteurs reviennent en détail sur les conditions d'existence d'une solution.

De même, dans (Ciampi, 1995), la fonction à minimiser lors de la phase d'apprentissage est explicitée. Celle-ci consiste en une erreur quadratique définie par l'équation (1.7).

$$E = \frac{1}{N} \sum_{i=1}^N Ei = \frac{1}{NJ} \sum_{i=1}^N \sum_{j=1}^J \left(Y_j^{(i)} - S_j^{(i)} \right)^2 \quad (1.7)$$

Dans cette équation, les notations suivantes sont utilisées :

- N , le nombre d'observations de la base d'apprentissage
- J , le nombre de neurones de la couche de sortie
- S , le vecteur des sorties calculées
- Y , le vecteur des sorties observées

Dans (Muller, 1993), les auteurs signalent un résultat intéressant, obtenu par Cybenko et Hornik : “Toute fonction continue sur un compact peut être approchée avec une précision arbitraire par un perceptron multicouche (ajustement non linéaire) comportant une seule couche cachée pourvue d'un nombre suffisant d'unités, et dont la fonction d'activation des neurones de la couche cachée est non-polynômiale, comme la fonction sigmoïde.”

Le deuxième argument donné en faveur des réseaux de neurones, dans cet article, est de dire que les modèles type ARMA ne faisant intervenir que des relations linéaires sont forcément inclus dans les réseaux neuronaux. Cela est facile à démontrer si l'on choisit comme fonction d'activation la fonction tangente hyperbolique qui a comme développement limité au voisinage de zéro $\tanh x = x + o(x^3)$. Ainsi, le raisonnement des auteurs consiste-t-il à dire que si la meilleure prévision pour une série temporelle est linéaire, elle sera trouvée par le perceptron multicouche.

Enfin dans (Canu, 1995), les auteurs citent Tong dans “Non-linear time series. A dynamical system approach” lorsqu'il donne la limite de l'utilisation des modèles ARMA : “La classe des modèles ARMA constitue un choix judicieux si et seulement si les autocovariances sont considérées comme une caractéristique importante du phénomène”. Dans cet article, les auteurs parlent des compétitions gagnées par les réseaux de neurones comme celle dite de “Santa Fe”.

Certains auteurs donnent des arguments liés au type de données utilisées pour justifier leur choix d'une modélisation par réseaux de neurones. Ainsi dans (Bigot, 1995), l'auteur justifie l'utilisation des réseaux de neurones en arguant de la multiplicité des erreurs possibles au niveau des données à utiliser. Ces erreurs sont dues aux capteurs, au modèle hydraulique et à la structure (prévision de consommation et mauvaise connaissance du réseau). L'auteur précise que les réseaux de neurones sont moins sensibles que les modèles statistiques classiques aux données bruitées, incomplètes ou, en partie, erronées. De plus la structure des algorithmes bâtis à partir de réseaux de neurones permet un haut degré de parallélisme donc une vitesse de résolution plus importante que les méthodes classiques.

1.1.3.3 Le nombre de couches et de neurones

La mise au point d'un modèle de prévision basé sur les réseaux de neurones suppose la détermination du détail de l'architecture de ceux-ci. C'est pourquoi la quasi-totalité des auteurs se pose la question du nombre de neurones en entrée et en sortie ainsi que du nombre de couches cachées (et du nombre de neurones dans ces couches cachées). Cette discussion est souvent délicate car chacun veut construire un modèle donnant des résultats proches des données existantes tout en évitant le sur-apprentissage lié à un nombre trop important de paramètres.

1.1.3.3.1 Le nombre de neurones d'entrée Dans (Canu, 1995), les auteurs cherchent à prédire la consommation d'eau potable du lendemain. Ils utilisent les données du tableau 1.3.

Type de paramètres	Nombre d'entrées
Pluviométrie des trois derniers jours	3
Température moyenne des trois derniers jours	3
Consommations constatées lors des sept derniers jours	7
Jour de la semaine	7
TOTAL	20

Tableau 1.3 : Choix des entrées dans le réseau de Canu

Il est à noter que l'auteur a choisi pour le jour de la semaine de faire intervenir sept neurones d'entrée prenant chacun la valeur 0 ou 1 à la place d'un neurone unique pouvant prendre sept valeurs.

Le choix de l'architecture du réseau, qui est fait ici, après un raisonnement empirique, est de compléter la couche d'entrée munie de 20 neurones et la couche de sortie ne comprenant qu'un neurone par une seule couche cachée comportant 4 neurones.

La particularité de la méthode utilisée dans (Wu, 1995) réside dans l'utilisation exclusive de variables endogènes en entrée du réseau de neurones. Le principe est donc celui d'une "fenêtre glissante". Les entrées et les sorties se correspondent comme le montre le tableau 1.4.

Avec n_i représentant le nombre de paramètres en entrée et n_0 le nombre de paramètres en sortie.

L'idée d'une telle méthode est de considérer que les variables exogènes, que l'on aurait tendance à introduire, sont implicitement intégrées dans les variables endogènes. Il est à noter au passage les intérêts majeurs d'un tel choix :

Entrées	Sorties
$X_1 \ X_2 \ X_3 \ \dots X_{n_i}$	$X_{n_i+1} \ X_{n_i+2} \ X_{n_i+3} \ \dots X_{n_i+n_0}$
$X_2 \ X_3 \ X_4 \ \dots X_{n_i+1}$	$X_{n_i+2} \ X_{n_i+3} \ X_{n_i+4} \ \dots X_{n_i+n_0+1}$
$X_{N-n_0-n_i+1} \ \dots X_{N-n_0}$	$X_{N-n_0+1} \ \dots X_N$

Tableau 1.4 : Choix des entrées et des sorties du réseau de Wu

- il n'est pas nécessaire de recueillir des informations météorologiques ;
- on évite ainsi de réaliser des prévisions de consommation à partir d'autres prévisions (les prévisions météorologiques).

Dans (Canu, 1990), les auteurs ont déterminé qu'il fallait utiliser en entrée les sept consommations précédentes. L'intérêt de ce chiffre est de reprendre la périodicité hebdomadaire du phénomène étudié. Les variables exogènes utilisées, en plus, en entrée sont la pluie et les températures des trois derniers jours ainsi que le jour de la semaine (sept neurones binaires).

Dans (Ding, 1995) la méthode utilisée pour déterminer le nombre de neurones en entrée est une méthode d'élimination. La procédure est initialisée avec le nombre maximum de paramètres (tels qu'ils peuvent être imaginés par l'auteur) puis ceux qui entraînent un poids trop faible sont éliminés progressivement.

1.1.3.3.2 Le nombre de réseaux de neurones Certains auteurs font le constat qu'il peut être moins efficace de vouloir construire un seul modèle pour réaliser des prévisions à tout moment de l'année ou de la semaine.

Dans (Sobral, 1991) l'application porte sur le réseau de la Région Parisienne Sud. Le modèle utilisé consiste en fait en deux réseaux de neurones pour chacune des rives de la Seine car leur industrialisation n'est pas du tout la même. Chacun de ces réseaux de neurones est constitué de deux couches cachées comportant respectivement 5 et 10 neurones. Les paramètres utilisés en entrée sont les suivants :

- la consommation pour les 7 jours précédents (7 neurones)
- la température et la pluviométrie des trois jours précédents (2*3 neurones)
- le jour de la semaine (7 neurones)

La sortie est composée d'un seul neurone donnant la prévision journalière.

Dans (Jain, 1994), les auteurs présentent dans un premier temps le réseau de neurones utilisé pour la prévision de la demande journalière de Lexington (Kentucky). Celui-ci comporte trois couches :

1. une couche d'entrée avec deux neurones (la demande du jour précédent et la température maximale prévue pour la journée),
2. une couche cachée avec trois neurones,
3. une couche de sortie avec un seul neurone (la demande de la journée à venir).

Par la suite, ils présentent une méthode qui mêle un système expert et les réseaux de neurones. L'aspect système expert consiste à définir trois domaines de prévision. A chacun de ces domaines, on associe un modèle de réseaux de neurones. Les trois modèles de réseaux de neurones ont la même structure (2 neurones en entrée, 3 dans la couche cachée et 1 neurone de sortie) ; la différence réside dans les fonctions d'activation mais surtout dans les variables des neurones d'entrée. Ces informations sont regroupées dans le tableau 1.5.

N°	Domaine	Paramètres en entrée
1	demande < seuil maximal	D_{t-1}
2	demande > seuil minimal et il ne pleut pas	D_{t-1} et T_t
3	demande > seuil minimal et il pleut	D_{t-1} et P_t

Tableau 1.5 : Réseaux de neurones de Jain

Le choix de ces trois domaines s'explique par une étude effectuée sur les consommations du site d'étude et dont les conclusions tenaient en trois points :

1. Lorsque la demande d'une journée est inférieure à un seuil donné, la demande du jour suivant est considérée comme constante.
2. Lorsque la demande d'une journée est supérieure à un seuil, qu'il n'a pas plu durant cette journée et qu'il ne pleut pas le jour de la prévision, la demande prévue est alors fonction de la demande de la veille et de la température prévue.
3. A l'inverse, lorsque la demande d'une journée est supérieure à un seuil mais qu'il a plu pendant cette journée ou/et qu'il pleut durant la journée sur laquelle la prévision est effectuée, la demande prévue est fonction de la demande de la veille et de la pluviométrie.

Dans (Muller, 1994) il s'agit de construire un modèle de prévision de la consommation électrique en France. Le pas de temps utilisé diffère des références rencontrées jusqu'alors puisqu'il s'agit de la demi-heure. Dans un premier temps l'idée est de réaliser cette prévision au moyen d'un seul réseau de neurones. Cette idée a été rejetée. Les auteurs trouvent, en effet, que les jours étant différents les uns des autres il est difficile de tout

regrouper dans un même outil. De plus, les prévisions devant se faire en boucle il est plus intéressant d'avoir des réseaux de neurones indépendants.

Les auteurs s'orientent alors vers la détermination d'une classification permettant de scinder les données utilisées en plusieurs sous-ensembles. La classification est réalisée à l'aide de la méthode des cartes auto-organisatrices de Kohonen. D'une utilisation classique pour les réseaux de neurones, cette méthode permet de dégager des regroupements homogènes parmi les données utilisées. On pourra en trouver les détails dans (El Ghaziri, 1993).

Nous pouvons retenir, pour l'application de (Muller, 1994), que deux séparations fortes existent. La première est hebdomadaire et permet de distinguer les week-ends et jours fériés d'une part et les autres jours de la semaine d'autre part. La seconde cassure est annuelle et permet de discerner trois saisons : Mai-Septembre, Octobre-Janvier et Février-Avril (la troisième étant nettement moins homogène que les deux autres du fait des variations de températures plus fortes au printemps).

Grâce à cette classification, nous arrivons à quatre modèles de réseaux de neurones : un pour le lundi, un pour la période mardi-vendredi, un pour le samedi (ainsi que les lundi et vendredi de week-ends fériés) et un pour le dimanche et les jours fériés. Chacun de ces modèles est construit sur la même logique. La saisonnalité constatée précédemment ne se traduit pas par la construction d'un réseau de neurones par saison mais par l'ajout de la variable saison en entrée de chacun des réseaux de neurones construits.

Les auteurs explicitent alors celui du mardi au vendredi. Il est composé de 48 réseaux de neurones (chacun pour la prévision d'une demi-heure donnée). Le neurone de sortie consiste en la différence entre la consommation du jour j au temps t et celle du jour $j - 1$ et du même temps t . Les entrées du réseau de neurones faisant la prévision de la demi-heure h sont :

- les 17 dernières valeurs de consommation (8 valeurs du type $c(d, h - i)$ pour $i=1$ à 8 et 9 valeurs du type $c(d - 1, h - j)$ pour $j=0$ à 8),
- les 17 dernières valeurs de la température ainsi que la prochaine valeur prévue,
- les 17 dernières valeurs de la nébulosité ainsi que la prochaine valeur prévue,
- les 5 valeurs des classes de saison,
- les 4 valeurs des classes des jours de la semaine (le mardi, le mercredi, le jeudi ou le vendredi).

De plus, il existe une couche cachée comportant 4 neurones. L'activation des neurones de la couche cachée se fait par une fonction sigmoïde alors que c'est une fonction linéaire

pour le neurone de sortie. Pour réduire le nombre de paramètres, on commence par utiliser une méthode de réduction des poids. Le résultat est de réduire le nombre de paramètres de 269 à une valeur comprise entre 114 et 207 suivant le pas de temps considéré.

Les résultats de ce type de modèle sont encourageants puisque sur le modèle exemple développé précédemment, le taux d'erreur relatif est compris entre 0,29 % et 0,60 %.

Les auteurs essayent pour finir un modèle en boucle où à un temps donné, on peut faire de la prévision pendant les 48 prochains pas de temps en utilisant les résultats de prévisions antérieures pour la nouvelle prévision. Dans ce cas l'erreur relative est comprise entre 0,03 % et 2,00 %.

1.1.3.3.3 Le nombre de couches cachées et de neurones par couche C'est au niveau des couches cachées que l'augmentation du nombre de paramètres des modèles peut rapidement devenir conséquente.

Dans (Mangeas, 1993) les auteurs proposent une structure originale de réseau de neurones afin de prédire la consommation d'électricité. La structure permet de tenir compte de la présence concomitante de variables endogènes, la demande d'électricité des pas de temps précédents notée X_{t-i} et exogènes. Ils proposent, en effet, de rajouter une couche cachée supplémentaire. Un exemple d'un tel réseau est représenté figure 1.1.

La première couche cachée permet, en fait, de réunir les variables endogènes et exogènes d'un pas de temps sur le même neurone. Ainsi, si l'on examine le pas de temps $t - 1$, on voit que les différents neurones d'entrée représentant les variables exogènes au pas de temps $t - 1$ sont réunis avec le neurone de la consommation à ce même pas de temps pour ne former en tout qu'un seul neurone dans la première couche cachée.

Il est également à noter sur ce schéma que les variables ε représentent un bruit aléatoire rajouté en entrée. Les fonctions d'activation mises en oeuvre figurent au-dessus de chaque couche. Il s'agit de fonctions sigmoïdes pour les deux couches cachées et d'une fonction linéaire pour la couche de sortie ne comportant qu'un seul neurone.

Dans (Bigot, 1995), l'auteur choisit de mêler, en entrée, des données endogènes et des données exogènes (relevés de pluviométrie et de température des trois jours précédents ainsi que jour de la semaine). La discussion sur le nombre de couches cachées et de neurones dans ces couches est ici particulièrement intéressante. L'auteur cite, en effet, les résultats d'autres auteurs comme Makhoul ou Cybenko qui ont prouvé que l'on pouvait approcher n'importe quelle fonction de classification et n'importe quelle fonction continue par un réseau de neurones à deux couches cachées. Toutefois, l'auteur remarque

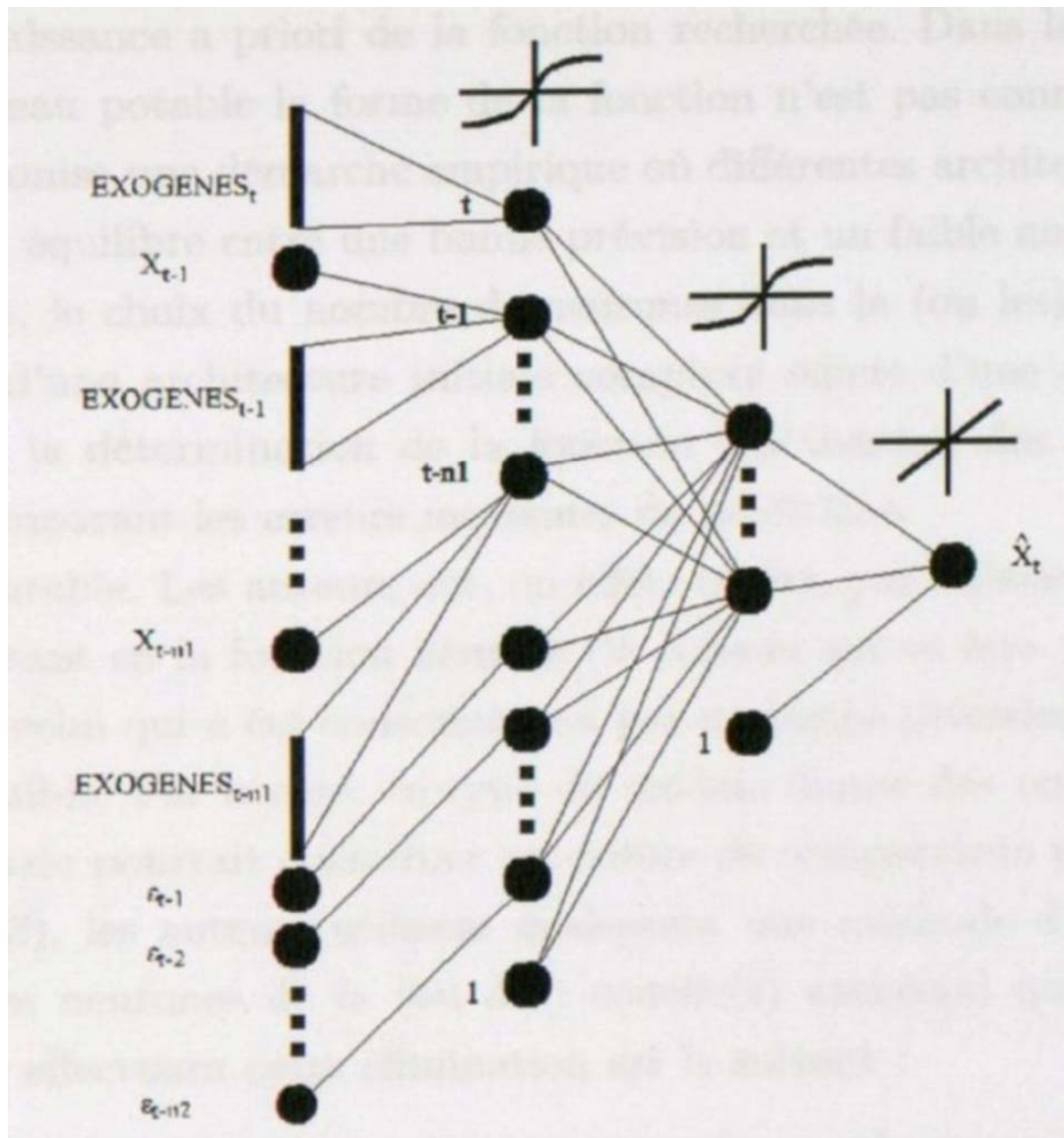


Figure 1.1 : Réseau de neurones proposé par Mangeas

qu'il peut être plus intéressant de construire un réseau avec trois couches cachées et peu de neurones par couche qu'un réseau avec deux couches cachées et davantage de neurones dans ces couches. En ce qui concerne le nombre de neurones par couche cachée, l'auteur signale qu'il serait possible de le déterminer si l'on avait une connaissance à priori de la fonction recherchée. Dans le cas de la prévision des consommations d'eau potable la forme de la fonction n'est pas connue à l'avance. C'est pourquoi l'auteur préconise une démarche empirique où différentes architectures seront testées afin de trouver un bon équilibre entre une bonne prévision et un faible nombre de paramètres.

Dans (Canu, 1990), le choix du nombre de neurones dans la (ou les) couche(s) cachée(s) s'effectue en partant d'une architecture initiale complexe suivie d'une simplification. Cette simplification, comme la détermination de la fonction d'activation des neurones, se fait de façon empirique en comparant les erreurs moyennes de prédiction.

Ce critère est discutable. Les auteurs ont, en effet, essayé, par ailleurs, un modèle de prévision simpliste consistant en la fonction identité (le volume qui va être consommé étant systématiquement égal à celui qui a été consommé au pas de temps précédent). Ce modèle donne une erreur moyenne faible. Par contre, ce type de modèle donne des erreurs maximales très fortes. L'erreur maximale pourrait constituer un critère de comparaison

plus pertinent.

Dans (Muller, 1993), les auteurs utilisent également une méthode d'élimination. Celle-ci concerne aussi bien les neurones de la (ou des) couche(s) cachée(s) que ceux de la couche d'entrée. L'algorithme effectuant cette élimination est le suivant :

1. On initialise les poids synaptiques aléatoirement (leurs valeurs ne doivent pas être trop grandes sous peine de saturer la totalité des fonctions sigmoïdes entraînant un blocage de l'apprentissage).
2. Apprentissage du réseau complet, selon une architecture initiale donnée : on calcule les $(n+1)p + p + 1 = (n+2)p + 1 = m$ poids minimisant l'erreur. Soit R le réseau trouvé.
3. On calcule pour tous les poids W_i , la valeur absolue des quotients $(Q_i)_{1 \leq i \leq m}$ de leur valeur estimée par leur écart-type (statistique du test $W_i = 0$ contre $W_i \neq 0$) avec $Q_i = \frac{W_i}{\sigma_i}$.
4. On détermine i tel que $|Q_i| = \min(|Q_j|)_{1 \leq j \leq m}$. Si de plus $|Q_i| < \beta$, β étant le fractile d'ordre $1 - \frac{\alpha}{2}$ de la loi normale, on élimine le poids correspondant, sinon on garde R . Soit R' le réseau R sans la connexion choisie.
5. Si le BIC du réseau R est inférieur à celui du réseau R' , on recommence en 2 à partir de R' , sinon on garde R .

Le critère BIC sera explicité dans le paragraphe 1.1.3.5.

1.1.3.3.4 Nombre de neurones de sortie Un modèle bâti à partir de réseaux de neurones peut permettre de prédire en même temps le volume qui sera consommé au cours de plusieurs pas de temps. Dans (Ding, 1995), les auteurs présentent différentes méthodes permettant de prédire grâce à un même réseau de neurones les différentes sorties $x(t), x(t+1), x(t+2), \dots$ et $x(t+\tau)$. Elles consistent à utiliser l'une des trois structures suivantes :

- des réseaux à sortie unique récurrents (la sortie du premier devient une entrée du second, etc...)
- un seul réseau de neurones à plusieurs sorties
- des réseaux (τ exactement) à sortie unique indépendants les uns des autres

C'est la dernière des trois solutions qui est retenue dans (Muller, 1994) pour la prévision de la consommation électrique par $\frac{1}{2}$ heure comme nous l'avons vu précédemment.

1.1.3.4 Les méthodes d'apprentissage

Une des caractéristiques essentielles d'un modèle à base de réseaux de neurones consiste en un apprentissage du phénomène à modéliser. Cet apprentissage se fait concrètement à partir d'un jeu de données. Comme nous l'avons déjà vu, cet apprentissage peut être supervisé ou non supervisé. Dans le premier cas les données nécessaires seront à la fois les entrées du système (suivant les exemples vus au chapitre précédent) et les sorties (la demande).

Les algorithmes utilisés pour cette phase d'apprentissage et la méthodologie d'organisation des données utilisées constituent les deux éléments qui nous ont intéressés dans nos recherches bibliographiques.

Dans (Sobral, 1991), la méthode d'apprentissage est une rétro-propagation du gradient. Cette méthode est souvent la base des algorithmes d'apprentissage. Elle consiste, à partir d'une fonction objectif telle que celle définie dans (Ciampi, 1995) et qui avait été reprise au paragraphe 1.1.3.2, à rechercher le minimum suivant un axe préférentiel qui est porté par le gradient de cette fonction. Dans (Sobral, 1991), la rétro-propagation du gradient s'effectue en deux passages sur le jeu de données : le premier est aléatoire alors que le second a pour objectif de favoriser les données les plus récentes.

Dans (Muller, 1993), les auteurs proposent la méthode de Brogen, Fletcher, Goldfarb et Schamo (BFGS) basée sur une approximation de rang deux de la matrice Hessienne.

La phase d'apprentissage est dans la plupart des cas réalisée lors de la construction du modèle. Ce modèle est, par la suite, mis en place au sein d'un outil plus global qui a comme fonction principale la gestion optimale d'un réseau. Plusieurs auteurs abordent la question de l'actualisation de l'apprentissage. La plupart d'entre eux concluent qu'il est nécessaire de recommencer la phase d'apprentissage à intervalles réguliers pour tenir compte des nouvelles informations accumulées.

Dans (Canu, 1990), l'apprentissage est refait tous les jours avant de débiter la prévision des consommations de la journée à venir. Cette méthode est discutable surtout si elle est utilisée sans pré-traitement des données car elle risque de donner beaucoup de poids à des événements qui peuvent être passagers.

Dans (Canu, 1995), les auteurs traitent de l'exemple de la prévision de consommation d'eau potable pour une journée. Pour ce faire, ils utilisent une chronique de consommation journalière de 1976 à 1992 sachant qu'ils possèdent la température moyenne et la pluviométrie sur ces journées. Comme cela est classiquement le cas avec les modèles neuronaux, les données de consommation ont été divisées en deux sous-ensembles, les années 1976-1982 servant pour l'apprentissage et les années 1983-1992 constituant l'ensemble

de test (appelé aussi ensemble de validation).

1.1.3.5 La qualité de la prévision

Comme nous l'avons déjà signalé, les outils construits à partir de réseaux de neurones bénéficient de moins de justifications théoriques que les modèles statistiques classiques. Il reste néanmoins indispensable de pouvoir quantifier la qualité de la prévision réalisée grâce à ces outils.

Ce sont ces évaluations de la qualité de prévision que nous allons exposer maintenant. Ce sera aussi l'occasion de préciser, pour les références où nous ne l'avons pas encore fait, les domaines sur lesquels la prévision est construite.

Dans (Canu, 1995) les auteurs commencent par l'examen des données dans le temps. Celui-ci fait apparaître une tendance globale (croissance de la consommation moyenne au cours des années) et des périodicités annuelles et hebdomadaires. Ces particularités pourraient être prédites par un modèle statistique linéaire comme par un modèle à base de réseaux de neurones.

Par contre, le lien entre les consommations journalières et la température n'est visiblement pas linéaire. Il semble que la fonction liant ces deux données est à effet de seuil. En effet, en dessous d'une température d'environ $15^{\circ}C$ la température n'a que peu d'influence sur la consommation journalière alors, qu'au-delà, la consommation semble être proportionnelle à la température moyenne. Ce type de relation non linéaire justifie l'emploi des réseaux de neurones. De plus, les auteurs affirment que si l'on peut accepter la modélisation gaussienne (si l'on admet que la loi conjointe de toute famille finie d'observations est gaussienne) alors les techniques de prédiction linéaires sont optimales. Pour vérifier cette hypothèse ils utilisent un test de normalité qui consiste à tracer la droite de Henry. Ils constatent alors que la distribution n'est pas gaussienne du fait de valeurs fortes qui sont plus importantes que les valeurs faibles. La consommation d'eau corrigée de la tendance globale n'est pas répartie symétriquement autour de sa moyenne.

Les auteurs comparent finalement les résultats obtenus par ce réseau de neurones à ce qu'ils auraient obtenu par un modèle linéaire en pré-traitant les données pour éliminer l'aspect température. L'erreur moyenne sur la période 1893-1992 baisse de 3,79 % à 3,42 %. Il est à noter au passage que l'utilisation d'un réseau de neurones ne demande pas de pré-traitement ce qui procure une plus grande souplesse d'utilisation.

Dans (Lek, 1996), le but est de prédire le débit d'un cours d'eau marocain. Le réseau de neurones utilisé est un réseau à trois couches. Trois critères de validation sont utilisés :

1. l'étude des erreurs (valeurs observées-valeurs estimées) : en s'assurant que leur

moyenne est statistiquement nulle et qu'elles sont stationnaires par rapport à la variable estimée en sortie.

2. le coefficient de corrélation linéaire R de Pearson entre le débit estimé et le débit observé.
3. le coefficient de corrélation de rangs CR de Spearman.

Les auteurs comparent également les valeurs du coefficient R obtenu par les réseaux de neurones avec celles obtenues par un modèle linéaire autorégressif avec des variables exogènes (pluies). On peut voir les résultats de cette comparaison dans le tableau 1.6.

Type de modèle	“ à l’instant ”	“ un pas ”	“ multi-pas ”
Réseaux de neurones	0,722	0,887	0,908
Autorégressif	0,562	0,771	0,784

Tableau 1.6: Comparaison des valeurs de R entre le modèle par réseaux de neurones et le modèle Autorégressif

Dans (Sobral, 1991), les résultats sont comparés avec un modèle statistique classique type ARMA. Les erreurs sont plus faibles en moyenne avec un modèle par réseaux de neurones et comportent surtout un écart type plus faible qui prouve que ce modèle entraîne moins d'erreurs lourdes.

En examinant de plus près les jours d'écart entre les deux modèles, on se rend compte que le modèle par réseaux de neurones est plus performant lorsqu'il s'agit de prévoir les consommations qui font intervenir des liens non linéaires avec les entrées (comme le lien entre la pluviométrie et la consommation).

Les jours où la prévision par réseaux de neurones n'est pas bonne sont les jours fériés et les jours de départ en vacances. Les auteurs prévoient de rajouter ultérieurement ces paramètres en entrée.

Dans (Mangeas, 1993), les auteurs comparent souvent la démarche par réseaux de neurones à celle des modèles type ARMAX (Auto-Regressive-Moving-Average with exogenous variables) qui consistent en une régression linéaire. Une des constatations de base qu'ils font est de dire que si la relation doit être linéaire, un modèle à base de réseaux de neurones permettra de déterminer cette relation. A l'inverse, un modèle linéaire ne pourra que partiellement approcher une relation non linéaire. Les deux autres qualités principales des réseaux de neurones recensées par les auteurs sont les suivantes :

- ils sont moins sensibles aux valeurs aberrantes (du fait notamment d'une partie constante dans la sigmoïde) ;

- ils demandent en général moins de valeurs pour réaliser une prévision de même qualité.

Par contre ils citent également un certain nombre de limites aux modèles utilisant les réseaux de neurones :

- la phase d'apprentissage a tendance à rester bloquée dans des minima locaux ;
- La sur-paramétrisation existe et, de façon plus générale, il est toujours difficile de trouver l'architecture du réseau de neurones à utiliser.

Les réseaux de neurones manquent d'estimateurs de la qualité de la prévision. Pour pallier ce problème, les auteurs proposent de reprendre l'estimateur *BIC* utilisé pour les modèles ARMA (AutoRegressive-Moving Average) et qui se présente sous la forme (1.8).

$$BIC = \text{Log} \hat{\sigma}^2 + m \frac{\lg T}{T} \quad (1.8)$$

Les variables T , m et $\hat{\sigma}^2$ représentent respectivement le nombre d'observations de la série, le nombre de paramètres ($m \leq p + q$) et une estimation de la variance résiduelle. L'adaptation à des réseaux neuronaux se fait en prenant pour m le nombre de poids synaptiques et pour T le nombre de vecteurs contenus dans la base d'apprentissage. Ce critère tient compte des performances du modèle en terme d'ajustement (valeur de la variance résiduelle) et du nombre de paramètres utilisés (nombre de poids synaptiques) pour un nombre donné d'observations de la base d'apprentissage. A l'aide de ce critère, on peut comparer les modèles et choisir celui qui répond le mieux au problème avec un faible nombre de paramètres.

La pertinence des différents modèles est testée sur la base de la consommation d'électricité journalière en France entre 1986 et 1991 (l'apprentissage se faisant sur la période 1986-1990 et la généralisation sur l'année 1991). Les trois modèles comparés sont d'une part le modèle type ARMA (régression linéaire) noté 1, le modèle par réseaux de neurones complet noté 2 et le modèle par réseaux de neurones simplifié noté 3. Le modèle 3 est obtenu par un algorithme d'élimination des synapses à poids faible.

On voit donc que les modèles à base de réseaux de neurones donnent de meilleurs résultats. De plus, la méthode d'élimination des liaisons de poids faibles donne satisfaction puisque le nombre de paramètres du modèle n°3 est le plus faible avec une erreur relative sur l'année test qui est, elle-aussi, la plus faible.

Nous avons vu au paragraphe 1.1.3.3 que dans (Jain, 1994), les auteurs ont construit un modèle de prévision de la consommation d'eau au pas de temps journalier. Les auteurs utilisent deux paramètres pour comparer les différents modèles qu'ils ont construits :

Modèle n°	Erreur relative sur la base d'apprentissage	Nombre de paramètres	BIC (calculé sur la base d'apprentissage)	Erreur relative sur l'année de validation 1991
	(en pour cent)			(en pour cent)
1	2,21	29	20,30	2,25
2	2,01	40	20,29	2,26
3	2,10	21	20,23	2,15

Tableau 1.7 : Comparaison de modèles ARMA et réseaux de neurones

- le coefficient AARE (Average Absolute Relative Error) qui s'exprime de façon assez classique par :

$$AARE = \frac{1}{N} \sum_{i=1}^n ABS(Err_i) \text{ avec } Err_t = \frac{DO_t - DF_t}{DO_t} \times 100 \quad (1.9)$$

Dans cette expression DO représente les données observées et DF les données prédites.

- Le coefficient de “seuils statistiques” qui fournit une estimation du nombre de prévisions d'erreur supérieure à un seuil donné. Par exemple, le seuil statistique pour un niveau de 10 % est obtenu par $TS_{10} = \frac{x_{10}}{N} \times 100$ où N est le nombre total d'erreurs de prévision d'un modèle et x_{10} le nombre d'erreurs supérieures à 10 %.

Ces deux coefficients ne mettent pas en avant le même type d'erreur. Le premier donne le même poids à toutes les erreurs alors que le deuxième ne prend en compte que celles qui dépassent le seuil retenu. C'est pourquoi les conclusions dégagées par les auteurs sont contrastées puisque suivant les années où les prévisions sont réalisées la comparaison des deux coefficients ne donne pas la même hiérarchisation.

On obtient ainsi les résultats du tableau 1.8.

Modèle	Coefficient	1988	1989	1990	1991	1992
Réseau de neurones 2-3-1	AARE	5,38	5,91	5,34	4,95	5,23
	TS ₁₀	13,46	14,56	13,74	11,26	14,29
Système expert n°1	AARE	5,21	6,15	4,82	5,41	5,57
	TS ₁₀	14,84	18,13	13,74	13,74	15,38
Système expert n°2	AARE	5,36	6,16	4,87	5,42	5,43
	TS ₁₀	15,38	18,13	13,46	12,64	15,93
Combiné expert/neurones	AARE	5,17	5,98	4,75	4,90	5,70
	TS ₁₀	14,56	16,48	12,36	9,07	16,48

Tableau 1.8 : Critères d'évaluation proposés par Jain

L'intérêt de (Crommelynck, 1991) repose dans le fait que les auteurs ont comparé sur un même réseau (celui du SEDIF en région parisienne) et avec les mêmes données

(consommations journalières de 1981 à 1989) la prévision par les méthodes classiques de régressions linéaires et la prévision par réseaux de neurones. L'outil utilisé sur ce réseau jusqu'alors était la régression linéaire. Les comparaisons faites entre les deux méthodes ne font pas apparaître de différences significatives. Les méthodes connexionnistes sont toujours meilleures mais de très peu. Les auteurs s'attendaient à des prévisions nettement meilleures par les méthodes connexionnistes. Pour expliquer ce résultat, ils avancent deux explications : les données ont été triées "à la main" avant de les utiliser ce qui a permis d'éliminer les données bruitées et les données utilisées portent sur une période très longue qui a fourni un échantillon suffisamment riche. Si l'une de ces deux hypothèses n'avait pas été vérifiée, les modèles par réseaux de neurones auraient sans doute été plus performants que les modèles par régression.

La conclusion de cet article fait transparaître les espoirs que formulent les auteurs dans l'utilisation des réseaux de neurones pour la prévision de la consommation de plus petits réseaux d'adduction d'eau potable (AEP). Ces espoirs sont justifiés par les points suivants :

- le comportement sur un réseau plus petit sera en principe moins lissé ;
- le comportement de tels réseaux serait rapidement changeant ou influencé par quelques facteurs principaux (stations balnéaires, réseau rural comportant une industrie très consommatrice,...) plutôt que par une multitude de facteurs de faible influence ;
- les données seraient de moins bonne qualité ou issues d'acquisitions automatiques sans contrôle de cohérence et de fiabilité ;
- les pas de temps seraient plus fins (modèles horaires) ;
- enfin et surtout l'apprentissage pourrait se poursuivre dans le temps (les méthodes permettant d'y parvenir ne sont pour l'instant pas avancées).

1.2 La gestion en temps réel des réseaux d'eau potable

Cette partie concerne à la fois les méthodes d'optimisation d'un réseau d'eau potable et la façon dont la gestion est mise en oeuvre sur le terrain. Bien que ce chapitre porte clairement sur la gestion des réseaux en AEP, il nous arrivera de citer des références dans d'autres domaines dans le cas où ces références pourraient compléter notre approche.

1.2.1 Objectifs de la gestion d'un réseau d'eau potable

Dans toute référence qui traite l'aspect pratique de la mise en oeuvre, il est intéressant de dégager l'objectif que les auteurs envisagent pour l'outil qu'ils conçoivent. Ainsi dans (Carpentier, 1993) les auteurs dégagent trois grands types de gestion envisageables :

- la gestion à court terme

Il s'agit ici de minimiser des coûts de pompage et de traitement de l'eau en respectant les contraintes physiques du réseau. Les auteurs précisent que les outils informatiques qui peuvent découler des modèles sont au départ conçus pour remplacer l'intervention humaine. Ils ajoutent que l'on peut envisager pour sécuriser le système la présence d'un opérateur qui filtrerait les commandes optimales à passer au système. Les deux autres enjeux qui peuvent exister dans la gestion des réseaux d'eau potable sont la détection précoce et la localisation des fuites d'une part et le calage des paramètres utiles à la modélisation du réseau d'autre part. Ces deux derniers aspects ne demandent peut-être pas un travail en temps réel ;

- la gestion à moyen terme

Cette catégorie concerne la gestion des ressources souterraines et le choix du meilleur tarif électrique ;

- la gestion à long terme

Il s'agit là de conception de réseau (ou d'extension) où il faut faire un choix optimal de canalisations, de réservoirs,...

C'est aussi dans cette catégorie que l'on peut faire entrer la gestion des bases des données qui constituent une mémoire du comportement du réseau très utile lors de la conception de nouveaux projets.

Nous allons détailler maintenant ces différents objectifs.

1.2.1.1 Minimisation des coûts de production

Nous ne reviendrons pas sur cet objectif qui est souvent le point de départ de tout travail sur la gestion automatisée d'un réseau.

Nous présenterons différentes fonctions objectif lors de l'étude de chacune des méthodes au paragraphe 1.2.2.

1.2.1.2 Minimisation des coûts d'investissement

Certains auteurs mêlent, dans la fonction objectif qu'ils utilisent, un terme de coût relatif au fonctionnement du réseau (coût de traitement et coût électrique de fonctionnement des pompes) ainsi qu'un coût d'investissement. Les outils qui sont alors mis en place n'ont pas comme but la gestion en temps réel d'un réseau mais le choix le plus économique entre plusieurs solutions techniques.

Dans (Ulanicki, 1993) le but est de choisir de façon optimale de nouvelles pompes sur un réseau existant.

Dans (Shamir, 1979) l'optimisation consiste à minimiser un critère de coût lors de la mise au point d'un projet d'aménagement. Les variables de décision sont l'emplacement des pompes, le type de pompes, le diamètre des canalisations et la position des réservoirs. Dans l'article, sont passés en revue tous les types de réseaux (ramifiés et maillés) et plusieurs méthodes d'optimisation (programmations dynamique, linéaire et non linéaire).

Dans (Brémond, 1987) l'auteur exprime analytiquement le coût du renforcement d'un réseau existant en fonction uniquement des charges hydrauliques. Dans une première étape, les charges optimales sont déterminées par une méthode de gradient projeté. Dans une deuxième étape, les coûts marginaux liés aux contraintes sur les charges et les demandes sont calculés. Ces coûts marginaux permettent alors d'orienter les choix de renforcement.

Le but de l'étude présentée dans (Peng, 1995) est de fournir à des gestionnaires de réseaux de chauffage à distance (CAD) des outils leur permettant de prévoir au mieux les extensions à mettre en oeuvre. Une partie de ce document est consacré à la présentation du programme d'optimisation qui dans la fonction coût fait intervenir à la fois les coûts d'investissement et les coûts de fonctionnement. La résolution de ce programme se fait par la méthode du recuit simulé.

1.2.1.3 Recherche de fuites

Nous avons déjà, dans la partie consacrée à la prévision de la demande, abordé cet objectif qui a été le support de la thèse (Bigot, 1995). Celle-ci a consisté à mettre au point un logiciel permettant de déceler toute anomalie dans le fonctionnement courant d'un réseau d'eau potable. Pour ce faire, on dégage deux axes : le premier consiste à calculer l'état normal attendu du réseau (à l'aide d'un modèle hydraulique et d'un modèle de prévision de consommation) le deuxième consiste à définir au mieux les points de mesure permettant de définir l'état réel du réseau. Le modèle de simulation hydraulique utilisé est PROGRES.

Dans (Pez, 1996), les auteurs relatent une expérience menée aux Etats Unis et qui consiste à combiner un système de télésurveillance à un logiciel de simulation hydraulique. L'objectif de l'outil, appelé Piccolo Real Time, est alors de tester des stratégies de fonctionnement du réseau en temps réel.

1.2.2 Les méthodes d'optimisation utilisées

Quel que soit l'objectif de chaque auteur, il est toujours confronté au problème du choix de la méthode qui va servir à l'optimisation. La plupart des outils que nous présenterons relèvent de la programmation mathématique. On pourra tout d'abord se référer à des ouvrages complets sur l'ensemble des outils de la recherche opérationnelle comme (Berge, 1983), (Nobert, 1995), (Sakarovitch, 1984a), (Sakarovitch, 1984b), (Minoux, 1983a) ou (Minoux, 1983b).

Dans (Gilbert, 1990) et (Shamir, 1979) on peut trouver une comparaison entre plusieurs de ces méthodes. Elle se décompose en deux volets : la capacité de la méthode à modéliser tout type de situations hydrauliques et la rapidité de calcul des solutions sur un ordinateur.

Dans (Lobbrecht, 1997), l'auteur compare plusieurs méthodes de la programmation mathématique pour la gestion sur un horizon allant de plusieurs heures à plusieurs jours de tout un système hydrographique incluant la demande humaine.

1.2.2.1 Les systèmes expert

Une première définition de cet outil peut être empruntée à (Matas, 1996) : “ La finalité de tels produits est de formaliser la connaissance acquise durant de longues années par des spécialistes d'un sujet, les experts du système, pour en faire un outil capable de répondre en lieu et place de plusieurs experts à des questions précises. Un système expert est donc composé d'une base de connaissances, d'une base de règles et d'un outil de réflexion appelé moteur d'inférence.

Il s'agit d'un outil qui tend à se développer. L'importance qui est donnée à cet outil est sans doute en partie liée aux erreurs qui ont pu être commises avec des modèles bâtis à partir d'autres outils moins proches d'un fonctionnement humain. Le passage d'une gestion manuelle d'un réseau à une gestion par système expert se fait bien plus en douceur que vers un système radicalement nouveau. Ceci est bien-sûr particulièrement le cas lorsque le système expert est construit à partir des règles issues de la gestion passée d'un réseau.

Certains réseaux sont d'ailleurs dans une phase intermédiaire où les informations sont recueillies de façon automatique mais le traitement des informations est surtout fait par des opérateurs. Dans (Pellegrini, 1995), le but est de gérer un réseau de grande taille, celui de la ville de Paris intra muros. Certains chiffres sont assez éloquentes :

- Chaque jour, sont effectués 80 télécommandes de démarrage ou d'arrêt de pompe et 190 télécommandes de vannes ;
- 600 capteurs fournissent en temps réel les données nécessaires pour le contrôle du chlore, le débit, la turbidité et les niveaux d'eau dans les réservoirs.

Ces informations permettent à des opérateurs appelés “dispatchers” de modifier le fonctionnement de certains organes (usines de production d'eau, vannes, pompes, etc.). En retour ce sont ces mêmes informations qui leur permettront de savoir si la commande est bien passée et si elle a eu l'effet attendu.

Dans (Charrat, 1995), l'auteur commence par décrire le réseau d'étude, celui de la Courly (Communauté Urbaine de Lyon). La télémessure rapatrie 4000 données TOR (Tout Ou Rien) et 2000 données analogiques. Dans cette thèse, l'auteur pose en particulier le problème de la validation des données rapatriées :

- Pour les données TOR, il existe des tests de cohérence qui invalident les données (ex. VANNE_OUVERTE et VANNE_FERMEE vraies toutes deux). En général les erreurs sur les informations TOR sont dues à l'installation.
- Pour les données analogiques, l'auteur affirme que l'erreur est plus faible si l'on horodate des changements significatifs plutôt que si l'on prend la donnée à un pas de temps constant.

Il existe dans le système prévu un fonctionnement semi-automatique appliqué aux exhaures qui consiste en un “dialogue” entre l'homme et la machine. L'intervention de l'opérateur permet de classer les pompes suivant ses critères, le superviseur fait alors le choix des pompes à mettre en marche en tenant compte de ce classement.

L'intérêt de cette thèse est aussi de donner des critères d'évaluation de la conduite d'un réseau d'eau potable. En fait, 10 critères regroupés en 3 familles ont été mis au point :

- Critères de sécurité :
 1. Autonomie minimale observée,
 2. Autonomie maximale observée,
 3. Temps où l'autonomie est inférieure à 5 heures,

- 4. Nombre de groupes utilisés.
- Critères de qualité :
 - 1. Taux de renouvellement de l'eau dans les réservoirs calculé en faisant la somme des entrées,
 - 2. Taux de renouvellement calculé par différence entre hauteur *max* et *min*,
 - 3. Nombre de démarrages des groupes,
 - 4. Heure de démarrage après 22 heures.
- Critères d'économie :
 - 1. Fonctionnement pendant les heures creuses,
 - 2. Fonctionnement pendant les heures pleines.

Dans (Lannuzel, 1990), l'auteur décrit le fonctionnement d'un système expert. Ce système a nécessité une phase de recueil du comportement d'un expert face à un problème à résoudre. Certains parallèles avec d'autres domaines sont abordés (diagnostic médical, partie d'échec, etc) mais le plus gros du travail porte sur l'expertise du site d'étude. Il se dégage des méthodes utilisées (entretiens avec les experts, observations du comportement d'un expert dans le cas d'une information volontairement tronquée, etc) des tendances sur le niveau de gravité et d'urgence de situations potentielles. Ces situations potentielles concernent le niveau bas dans les différents réservoirs, le dépassement de la puissance maximale fournie par le contrat EDF, la pression en sortie de station et le nombre de changements d'état pour les pompes.

Une partie importante de (Darnault, 1991) est consacrée à l'étude d'un incident généralisé sur un réseau électrique. La situation de reprise relève de problèmes multi-objectifs et l'auteur utilise alors un système expert.

1.2.2.2 La programmation dynamique discrète

Le principe de la méthode est toujours le même. Il consiste dans un premier temps à décomposer les variables d'état et de décision en pas discrets. Dans un deuxième temps il s'agit de faire varier les variables de décision afin d'examiner tous les états admissibles et de retenir celle qui minimise la fonction objectif à chaque pas de temps.

Pour présenter la façon dont on traite un cas réel, on peut reprendre celui qui avait été traité dans (Guhl, 1995) et qui concernait un réseau d'eau potable simple comportant un réservoir et une pompe. Nous avons alors décomposé le volume du réservoir de $1\,000\,m^3$

en dix pas de volume de 100 m^3 . Avec les volumes de consommation qui avait été pris pour les six pas de temps étudiés, nous obtenions le cheminement tracé sur la figure 1.2.

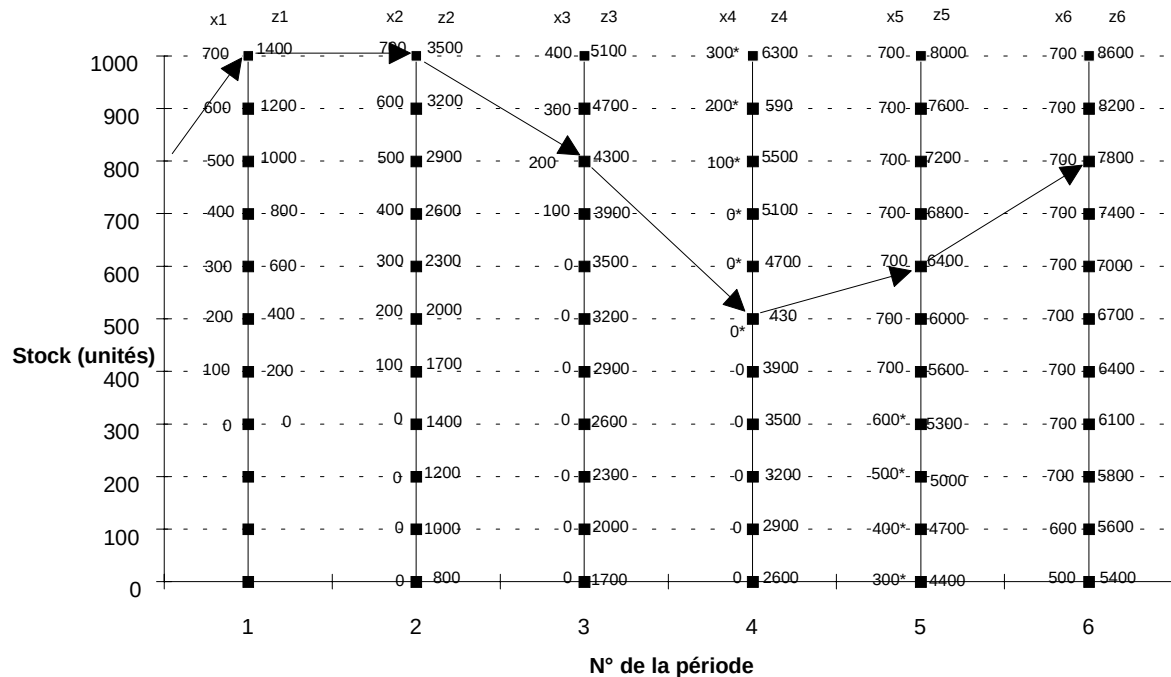


Figure 1.2 : Exemple de résolution par la programmation dynamique

Dans le graphique de la figure 1.2 la variable x représente la valeur de la commande et z la valeur de la fonction coût optimal pour chacune des valeurs de l'état du stock. Lorsque le chiffre de la commande est suivi d'un astérisque cela signifie qu'il existe plusieurs solutions à coût équivalent permettant de passer d'une période à la suivante, le chiffre alors fourni est l'une des solutions.

La taille du réseau qui sert ici d'exemple est très restreinte. Il est assez significatif de ce qu'il est possible de traiter par cette méthode.

Il est notamment possible de voir dans (Amoretti, 1996) un exemple d'optimisation d'un réseau par la Programmation Dynamique. Dans cet article le réseau d'étude théorique comporte un réservoir et une pompe. Des résultats de simulation sont montrés. Ils font apparaître l'intérêt d'une optimisation globale. En effet, dans tous les cas le coût d'une journée optimisée est moins fort que le coût d'une journée menée par une méthode traditionnelle (niveaux d'arrêt et de démarrage fixes dans les réservoirs). De plus, le nombre de situations critiques (passage au radier) est plus faible. Plus les volumes distribués sont importants ou plus l'erreur sur la prévision de consommation est forte

plus l'optimisation est intéressante.

C'est le même type de réseau test qui est utilisé dans (Flowers, 1991) où l'auteur présente une étude qui a abouti à la réalisation d'un programme informatique : ESEM (Expert System for Energy Management). Ce programme tourne sous Windows et comporte 3 parties : la prévision de consommation, l'optimisation des stratégies de pompages réalisée par la programmation dynamique et une base de données du système. L'exemple présenté pour prouver l'efficacité de ce programme comprend une pompe et un réservoir.

Dans (Renaud, 1993) et (Carpentier, 1993) la programmation dynamique est utilisée pour modéliser le phénomène. Comme cette méthode est couplée avec une méthode de décomposition-coordination, nous verrons ces applications dans le paragraphe consacré à cette deuxième méthode.

Dans (Lansey, 1994) et (Brion, 1991), le même problème est présenté puisqu'il s'agit de l'optimisation économique du fonctionnement du réseau d'Austin, capitale de l'État du Texas. Plus précisément, ces études ont comme but de tenir compte du nombre d'arrêts/démarrages de pompes dans l'optimisation de la gestion d'un réseau d'eau potable.

La formulation du problème se fait toujours suivant la programmation non-linéaire de façon semblable à ce que nous avons vu précédemment. L'utilisation des méthodes proposées devant pouvoir se faire en temps réel, les auteurs proposent, pour la résolution de ce programme, l'utilisation d'une technique en deux temps.

Dans un premier temps ils utilisent une pré-optimisation. Celle-ci consiste en une simulation sur modèle hydraulique (le modèle utilisé ici s'appelle KYPIPE) permettant d'obtenir des courbes liant la permutation des pompes à la hauteur d'eau dans le réservoir amont ainsi que l'énergie absorbée par la pompe toujours en fonction de ce niveau amont. A partir de ces courbes, des régressions polynômiales sont effectuées. Celles-ci permettent, a priori, de gagner du temps dans la phase d'optimisation car il est alors possible de calculer le coût de fonctionnement d'une pompe à tout moment.

La deuxième étape d'optimisation est une résolution par Programmation Dynamique Discrète. Malgré la pré-optimisation, le temps de calcul semble être prohibitif puisque l'essai est fait sur un réseau ne comportant qu'un seul réservoir. Les auteurs concluent que, pour un réseau plus complet, on pourrait opérer un découpage en sous-réseaux homogènes (en terme de pression).

1.2.2.3 La programmation non linéaire

Dans la comparaison des différentes méthodes faite dans (Gilbert, 1990), l'auteur signalait déjà que certains réseaux ne pouvaient pas être modélisés par des méthodes

linéaires. Dans cette thèse, l'auteur parlait notamment d'une situation hydraulique classique où deux réservoirs se trouvent en équilibre hydraulique.

Pour pouvoir être davantage proche de la réalité physique que l'on modélise, certains auteurs ont choisi d'utiliser plutôt la programmation non linéaire.

Dans (Ormsbee, 1994), la non-linéarité vient du fait que les auteurs traitent tout l'horizon d'optimisation en une seule écriture. Dans ce cas, on rencontre des fonctions coûts électriques qui sont constantes par morceaux et donc non linéaires.

Dans (Yu, 1994), les auteurs justifient l'emploi de la programmation non linéaire pour optimiser un réseau d'eau potable. Les auteurs rejettent tous les modèles qui ont choisi de linéariser le problème pour pouvoir utiliser la programmation linéaire. Ils considèrent que cette simplification est trop forte et que le problème est trop peu linéaire. Ils rejettent également la programmation dynamique en la considérant trop gourmande en temps de calcul. Enfin, ils rejettent la programmation dynamique couplée avec une technique de décomposition-coordination car ils pensent que cette technique ne converge pas forcément.

Les auteurs développent un exemple fictif qui comporte les éléments suivants :

- deux stations de pompes servant de ressource au système,
- deux réservoirs munis chacun d'une vanne en sortie,
- 48 noeuds de consommation (affectés d'une courbe de consommation fictive).

En fait on s'aperçoit dans cet exemple que l'optimisation qui est proposée est celle d'un réseau de distribution. La résolution du problème se fait par une descente de gradient généralisée.

Enfin, les temps de calcul obtenus dans cette étude confirment la réticence que l'on peut avoir à utiliser de telles méthodes pour la gestion en temps réel de réseaux. Pour cet exemple comportant très peu de variables de décision (4) et de contrôle (2), on atteint un temps de 4 mn et 42 sec sur un VAX-310.

Dans (Pezeshk, 1994) l'auteur traite de la minimisation des coûts d'énergie de pompage inhérents à la production d'eau dans un forage. La comparaison de plusieurs méthodes de minimisation des coûts est effectuée. Cette minimisation est obtenue en combinant un certain nombre de pompes.

Ce qui est intéressant ici est la forme de la fonction objectif qui comporte un terme classique $\sum_{i=1}^I \frac{\gamma TDH_i Q_i}{e_i}$ et un terme de pénalité. Dans l'équation précédente, les variables suivantes sont utilisées :

- I , le nombre de pompes dans le réseau

- Q_i , le débit de la pompe i
- e_i , le rendement de la pompe i
- γ , le poids spécifique de l'eau
- TDH_i (Total Dynamic Head), la hauteur manométrique totale fournie par la pompe i

De plus, nous avons également les relations habituelles entre TDH_i et Q_i ainsi qu'entre e_i et Q_i .

$$TDH_i = A_i^{tdh} Q_i^2 + B_i^{tdh} Q_i + C_i^{tdh} \quad (1.10)$$

$$e_i = A_i^e Q_i^2 + B_i^e Q_i + C_i^e \quad (1.11)$$

Dans (Ulanicki, 1991), on présente également en détail l'écriture de la fonction objectif du programme non linéaire. Celle-ci prend la forme (1.12).

$$J_T = \sum_{k=0}^{K-1} \sum_{i=1}^I J_i [x(k), u(k), d(k)] + \sum_{k=0}^{K-1} C q_0(k) \quad (1.12)$$

Dans cette relation, nous utilisons les notations suivantes :

- I : nombre de stations de pompage sur le réseau,
- K : nombre de pas de temps sur lequel a lieu l'optimisation,
- q : vecteur des flots dans le réseau,
- q_0 : vecteur particulier correspondant au débit d'eau entrant dans les stations de traitement,
- C : matrice des coûts de traitement aux usines,
- d : vecteur des demandes aux réservoirs,
- x : vecteur des niveaux de réservoir,
- u : vecteur des commandes de pompes et de vannes.

La première partie de cette expression J_i peut ainsi ne pas être linéaire, elle représente le coût de fonctionnement de chaque station de pompage.

Dans (Amoretti, 1990), la non-linéarité vient de la façon de construire le modèle. En effet, le modèle utilisé pour l'optimisation des coûts proportionnels sur un réseau AEP (celui du FIUM'ORBO en Corse) est un modèle conceptuel. Il s'écrit de la façon suivante :

$$H(t + dt) = H(t) + S(t) + A.P(t) + D.C(t) \quad (1.13)$$

avec les notations suivantes :

- $H(t)$, le vecteur des n volumes d'eau au temps t ,
- $S(t)$, le vecteur de l'apport des sources dans chaque réservoir,
- $P(t)$, le vecteur des m pompages entre t et $t + dt$.
- A , une matrice ($n \times m$) dont l'élément a_{ij} représente la contribution du pompage j au réservoir i ,
- $C(t)$, la consommation globale de t à $t + dt$,
- D , un vecteur (de dimension n) dont chaque élément d_i représente la fraction de consommation $C(t)$ fournie par le réservoir i .

Deux phénomènes ne sont pas pris en compte dans un tel modèle. D'une part, il peut y avoir une sortie d'eau dans un réservoir en l'absence de pompage et de consommation du seul fait d'un déséquilibre entre deux réservoirs. D'autre part, l'écoulement dans les tronçons en régime permanent dépend de l'intensité des consommations et de leur localisation. Si l'on tient compte de ces éléments, le modèle devient :

$$H(t + dt) = H(t) + S(t) + [A + B.C(t)].P(t) + [D + E.C(t)].C(t) + F \quad (1.14)$$

avec cette fois :

- F , un vecteur (de dimension n) dont le terme f_i représente le débit constant au réservoir i .
- $B.C(t)$, une matrice ($n \times m$) dont l'élément $b_{ij}.C(t)$ représente un terme correctif à la contribution a_{ij} du pompage j au réservoir i .
- $E.C(t)$, un vecteur (de dimension n) dont l'élément $e_i.C(t)$ représente un terme correctif à la fraction d_i de la consommation $C(t)$ fournie par le réservoir i .

Le fait d'avoir ainsi voulu regrouper tous les pompages en un seul vecteur, séparé ensuite par une matrice attribuant à chaque réservoir l'incidence de chaque pompage, rend le modèle non linéaire. C'est pourquoi cette méthode amène à une résolution par la programmation non linéaire. Plus précisément la résolution se fait par l'algorithme du gradient conjugué.

Un autre inconvénient de cette méthode est le nombre de paramètres à caler dans le modèle qui atteint tout calcul développé la valeur $N = (n - 1) \times (2m + 3)$ avec n le nombre de réservoirs et m le nombre de station de pompage.

Enfin le temps de calcul d'une telle méthode est très important puisque le tableau de résultat montre qu'il peut être de 15 à 30 minutes pour 5 réservoirs et 3 stations de pompage.

Dans (Carpentier, 1993), l'ensemble des équations linéaires et non linéaires permettant à tout instant de lier les pressions aux débits est explicité. La résolution de ces équations est réalisable par une méthode de Newton par exemple.

L'équation donnant le coût de fonctionnement d'un réseau en fonction du débit transitant est explicitée. Cette équation fait apparaître que les coûts liés au traitement sont proportionnels au débit alors que les coûts de pompage ne sont pas linéaires puisqu'ils s'expriment par la relation (1.15).

$$c_{ij} = \beta_{ij} \frac{\pi_{ij} q_{ij}}{\rho_{ij}(q_{ij})} \quad (1.15)$$

Avec β qui représente le coût unitaire de la puissance électrique de la pompe, π la hauteur d'élévation de la pompe et ρ le rendement de la pompe dépendant du débit q_{ij} .

Dans (Cembrano, 1988), le réseau étudié est celui de la ville de Barcelone. L'aspect intéressant (et surtout différent des autres auteurs) est l'écriture de la fonction coût (ou fonction objectif dans le sens de l'optimisation). Celle-ci comprend une partie linéaire (par rapport au flot) qui correspond aux coûts de traitement.

Le coût de transit des flots dans les pompes et les vannes est lui composé de trois termes. Un terme linéaire et un terme cubique correspondent aux coûts de pompages nécessaires d'une part à l'élévation théorique et d'autre part à la perte de charge due aux frictions. Le troisième terme est en carré (de la différence entre le flot de référence correspondant à la pression optimale attendue par les usagers) créant une pénalité inhérente à la mauvaise adéquation entre les attentes des usagers et le service réel.

La résolution s'opère en effectuant une descente de gradient.

Enfin, nous pouvons, pour terminer ce paragraphe citer d'autres références utilisant la programmation non linéaire de façon assez analogue à ce que nous avons vu jusqu'ici. Il sera ainsi intéressant de se référer à (Sterling, 1975) ou (Shamir, 1974), où la méthode utilisée est toujours la programmation non linéaire, la résolution se faisant alors par l'algorithme de Newton-Raphson.

1.2.2.4 La programmation linéaire

A condition de simplifier en partie le problème étudié, il est possible de présenter l'optimisation du fonctionnement d'un réseau d'eau potable sous la forme d'un programme linéaire. La linéarité de la fonction objectif est souvent due au fait que l'on décompose le problème général par pas de temps (d'une durée d'une heure la plupart du temps). Ceci évite le problème de la non-linéarité des coûts d'électricité sur 24 heures.

L'ouvrage (Chvatal, 1983) présente l'ensemble des méthodes et des algorithmes relevant de la programmation linéaire.

On pourra se reporter à (Bryds, 1992) pour voir le détail d'un programme linéaire visant l'optimisation d'un réseau d'AEP sur un horizon de 24 heures. La résolution de ce programme se fait par relaxation lagrangienne.

Cette méthode est également utilisée dans (Crawley, 1993) où les auteurs présentent l'optimisation de toute une partie du New South Wales en Australie. La zone considérée (la région d'Adelaide) représente l'alimentation en eau de quelque 1 000 000 d'habitants.

On peut aussi, à condition de simplifier l'écriture du problème, mettre en oeuvre la simulation du fonctionnement d'un réseau d'eau potable en terme d'optimisation par programmation linéaire. C'est ce qui est proposé dans (Berghout, 1997). Une amélioration de ces méthodes, appelée "feedback devices" permet de tenir compte de réducteurs de pression et de pompes à vitesse variable.

L'algorithme le plus courant de résolution d'un programme linéaire est celui du Simplexe. MESZAROS compare cette méthode à celle du point intérieur dans (Mészáros, 1996). Cette dernière méthode qui connaît un regain d'intérêt depuis le début des années 80 donne des résultats plus intéressants en terme de rapidité de convergence des calculs.

Une partie importante de ce document sera consacrée à l'utilisation de la programmation linéaire pour l'optimisation des réseaux d'eau potable. C'est la raison pour laquelle nous présentons dans les paragraphes suivants la bibliographie consacrée à des points particuliers de la programmation linéaire.

1.2.2.4.1 La programmation linéaire en nombres mixtes Une technique proche de la programmation linéaire est la programmation mixte où certaines variables sont astreintes à être entières. C'est cette technique qui est utilisée dans (Ulanicki, 1993) où les variables de décision d'investissement (construction d'un ouvrage ou d'une station de pompage) peuvent prendre la valeur 0 ou 1.

Un des chapitres de (Golstein, 1973) est consacré à la présentation de problèmes relevant de la programmation linéaire (partiellement ou totalement) en nombres entiers. Les auteurs distinguent trois grands domaines d'application :

- phénomènes portant sur des objets indivisibles : usines, wagons, avions, etc,
- les problèmes de type combinatoire (tel le problème du voyageur de commerce ou l'ordonnancement de tâches),
- les problèmes non linéaires qui dépassent le cadre de la programmation convexe.

En particulier, ils présentent l'utilisation de programme en nombres entiers pour résoudre des programmes non linéaires. Ils précisent qu'il est utile d'utiliser cette méthode lorsque le problème non linéaire sort du cadre restrictif de la programmation convexe. Cette dernière répond à deux critères : la fonction objectif est convexe ou concave et l'ensemble des contraintes forme un domaine convexe.

L'exemple support de la transformation d'un programme non linéaire en un programme en nombre entier utilisé dans (Golstein, 1973) est dit de distribution. Nous pouvons voir en détail comment les auteurs transforment le programme (1.16).

$$\left\{ \begin{array}{l} \text{Min } F(X) = \sum_{i=1}^m \sum_{j=1}^n \left[c_{ij} + \varphi_i \left(\sum_{j=1}^n x_{ij} \right) \right] x_{ij} \\ \text{sous les contraintes} \\ \forall j \in \langle 1; n \rangle, \sum_{i=1}^m x_{ij} = b_j \\ \forall i \in \langle 1; m \rangle, \forall j \in \langle 1; n \rangle, x_{ij} \geq 0 \end{array} \right. \quad (1.16)$$

Dans ce programme (1.16) x_{ij} représente la quantité transportée de A_i à B_j . La fonction φ explicite la relation entre le prix de revient d'une unité de produit et le volume de fabrication (a priori cette fonction doit être décroissante puisqu'elle permet de tenir compte de l'amortissement des frais d'investissement).

Pour transformer ce problème en programme linéaire en nombre entier, on introduit une hypothèse supplémentaire. Celle-ci consiste à dire que la production ne prend que des valeurs discrètes. On suppose que les valeurs possibles du volume de production en A_i (qui est noté X_i et qui s'exprime par $X_i = \sum_{j=1}^n x_{ij}$) sont $d_{\lambda i}$ avec $\lambda \in \langle 0; s_i \rangle$. On convient d'avoir :

$$d_{0i} = 0 < d_{1i} < \dots < d_{s_i i}$$

On peut alors écrire : $X_i = \sum_{\lambda=1}^{s_i} (d_{\lambda i} - d_{\lambda-1,i}) y_{\lambda i}$ en utilisant des variables $y_{\lambda i}$ astreintes à valoir 0 ou 1. Plus précisément, les variables $y_{\lambda i}$ doivent vérifier les propriétés suivantes :

$$\left\{ \begin{array}{l} y_{1i} = y_{2i} = \dots = y_{\lambda_0 i} = 1 \\ y_{\lambda_0+1,i} = y_{\lambda_0+2,i} = \dots = y_{s_i i} = 0 \end{array} \right.$$

$$\text{Ceci peut s'obtenir par le système : } \left\{ \begin{array}{l} \forall \lambda \in \langle 2; s_i \rangle, y_{\lambda-1,i} \geq y_{\lambda i} \\ \forall \lambda \in \langle 1; s_i \rangle, y_{\lambda i} \in \{0; 1\} \end{array} \right.$$

Pour rendre toute la fonction initiale linéaire, il est nécessaire d'introduire les notations suivantes : $\psi_{\lambda i} = d_{\lambda i} \varphi_i(d_{\lambda i})$. On peut exprimer les produits en utilisant à nouveau les variables $y_{\lambda i}$ en écrivant : $X_i \varphi_i(X_i) = \sum_{\lambda=1}^{s_i} (\psi_{\lambda i} - \psi_{\lambda-1,i}) y_{\lambda i}$. Le programme (P)

précédent s'écrit sous la forme .

$$\left\{ \begin{array}{l} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{i=1}^m \sum_{\lambda=1}^{s_i} (\Psi_{\lambda i} - \Psi_{\lambda-1,i}) y_{\lambda i} \\ \text{sous les conditions} \\ \forall j \in \langle 1; n \rangle, \sum_{i=1}^m x_{ij} = b_j \\ \forall i \in \langle 1; m \rangle, \sum_{j=1}^n x_{ij} \leq \sum_{\lambda=1}^{s_i} (d_{\lambda i} - d_{\lambda-1,i}) y_{\lambda i} \\ \forall i \in \langle 1; m \rangle, \forall j \in \langle 1; n \rangle, x_{ij} \geq 0 \\ \forall i \in \langle 1; m \rangle, \forall \lambda \in \langle 1; s_i \rangle, y_{\lambda i} \in \langle 0; 1 \rangle \\ \forall i \in \langle 1; m \rangle, \forall \lambda \in \langle 1; s_i \rangle, y_{\lambda-1,i} \geq y_{\lambda i} \end{array} \right. \quad (1.17)$$

Ils consacrent, ensuite, un paragraphe à l'étude des conditions pour qu'un programme linéaire donne directement des résultats entiers sans qu'il soit nécessaire d'astreindre les inconnues à être entières. Enfin, ils exposent une méthode de résolution d'un programme où toutes les variables sont astreintes à être entières. La méthode consiste à résoudre à l'aide du simplexe, puis à rajouter des contraintes permettant d'exclure la solution non entière trouvée puis à résoudre à nouveau et ainsi de suite. Ils adaptent la méthode précédente pour répondre à des programmes où seulement certaines variables sont astreintes à être entières.

L'algorithme de résolution le plus souvent utilisé en programmation linéaire en nombres mixtes est du type séparation-évaluation (branch and bound). La méthode est expliquée en détail dans (Minoux, 1983).

Le principe de la méthode consiste à parcourir un arbre où les différents noeuds correspondent à des contraintes d'intégrité sur une partie des variables initialement laissées libres. Nous reviendrons au paragraphe 5.3 sur la méthode de branch and bound et sur les aspects algorithmiques que présente Minoux.

1.2.2.4.2 La programmation linéaire successive Cette méthode appelée aussi SLP consiste à résoudre un programme non-linéaire en s'affranchissant des méthodes de la programmation non-linéaire. Les algorithmes présentés consistent donc en une linéarisation du programme à chaque pas pour pouvoir utiliser la programmation linéaire.

Minoux consacre un chapitre de (Minoux, 1983) à ce que l'auteur appelle les méthodes de linéarisation qui consistent de la même façon à remplacer le problème initial par une succession de problèmes linéaires. L'auteur distingue deux grandes familles de méthodes :

- l'approximation tangentielle où une fonction non linéaire continûment différen-

tiabile est remplacée au voisinage de $x = x^0$ par la fonction affine (1.18).

$$z = h(x^0) + \nabla h^T(x^0) \cdot (x - x^0) \quad (1.18)$$

Cette méthode est proche de celle décrite dans (Palacios-Gomez, 1982).

- L'approximation barycentrique où pour une fonction non linéaire convexe $h(x)$ on se donne K points $y^1, y^2, \dots, y^K$ qui nous permettent de remplacer la fonction $h(x)$ par le programme (1.19).

$$\begin{cases} h(x) = \text{Min} \left\{ \sum_{k=1}^K \lambda_k h(y^k) \right\} \text{ sous les contraintes :} \\ \sum_{k=1}^K \lambda_k y^k = x \end{cases} \quad (1.19)$$

Si la fonction h est concave, le *Min* est remplacé par un *Max* dans (1.19).

L'auteur présente alors la méthode de Frank et Wolfe qui permet de traiter des problèmes de la forme :

$$\begin{cases} \text{Min } f(x) \\ \text{sous les contraintes} \\ Ax = b \\ x \geq 0 \end{cases}$$

La méthode est itérative et engendre une suite de points $x^0, x^1, \dots, x^k$ où x^{k+1} est déterminé à partir de x^k de la façon suivante :

On commence par résoudre le programme linéaire suivant $PL(x^k)$

$$\begin{cases} \text{Min } \nabla f^T(x^k) \cdot x \\ \text{Sous les contraintes} \\ Ax = b \\ x \geq 0 \end{cases} \quad (1.20)$$

Soit y^k , un point extrême solution de $PL(x^k)$. Alors x^{k+1} est choisi de façon à minimiser f sur le segment $[x^k, y^k]$. La convergence de cette méthode est assurée par le théorème suivant :

On suppose que f est continûment différentiable et que l'une des deux conditions suivantes est vérifiée :

- (i) le polyèdre $X = \{x / Ax = b, x \geq 0\}$ est borné
- (ii) $f(x) \rightarrow +\infty$ quand $\|x\| \rightarrow +\infty$

Alors pour tout point de départ $x^0 \in X$, la méthode de Frank et Wolfe converge vers un optimum local du problème $\text{Min}_{x \in X} f(x)$.

La deuxième méthode présentée est celle des plans sécants de Kelley qui s'applique à des problèmes convexes et s'apparente à la méthode des coupes en programmation linéaire en nombres mixtes.

La troisième méthode est l'algorithme de génération de colonnes de Dantzig qui est du type approximation barycentrique.

Dans (Lobbrecht, 1997), la méthode qui est présentée est un système de boucle optimisation-simulation : après chaque pas d'optimisation on réalise une simulation à l'aide d'un modèle pour calculer les nouvelles valeurs des inconnues et ainsi de suite. Malheureusement l'auteur ne dit pas grand chose de la convergence d'une telle méthode.

Dans (Palacios-Gomez, 1982), les auteurs commencent par présenter le programme non linéaire (1.21) qu'ils vont résoudre par cette méthode.

$$\left\{ \begin{array}{l} \text{Min } g_0(x) + a_0 y = h(x, y) \\ \text{sous les conditions} \\ g(x) + A_1 y = b_1 \quad (m_1 \text{ lignes}) \\ A_2 x + A_3 y = b_2 \quad (m_2 \text{ lignes}) \\ l_1 \leq x \leq u_1, l_2 \leq y \leq u_2 \end{array} \right. \quad (1.21)$$

Ils ont choisi de séparer les variables qui interviennent linéairement (comme les y) et les autres. A partir de là ils choisissent une valeur de x (notée $\bar{x}$) et écrivent le programme linéaire (1.22).

$$LP(\bar{x}, s) \left\{ \begin{array}{l} \text{Min } \nabla g_0(\bar{x}) d + a_0 y \\ \text{sous les conditions} \\ J(\bar{x}) d + A_1 y = b_1 - g(\bar{x}) \\ A_2 d + A_3 y = b_2 - A_2 \bar{x} \\ \max(l_1 - \bar{x}, -s) \leq d \leq \min(u_1 - \bar{x}, s) \\ l_2 \leq y \leq u_2 \end{array} \right. \quad (1.22)$$

Le programme (1.22) est alors résolu donnant une solution $(\bar{d}, \bar{y})$. Si le candidat successeur $(\bar{x} + \bar{d}, \bar{y})$ vérifie alors certains critères, il est accepté et la valeur de s est augmentée. Dans le cas contraire, s est réduit à la valeur s' et c'est $LP(\bar{x}, s')$ qui est résolu.

Dans (Baker, 1985), on peut voir la mise en oeuvre complète de la méthode SLP sur un exemple. Cet exemple est celui de l'optimisation de la production d'une raffinerie de pétrole. L'algorithme présenté ici diffère néanmoins un peu de celui de l'article cité précédemment. Les auteurs mettent, en particulier, en place des contrôles pour éviter que l'algorithme oscille sans converger.

Dans (Ferris, 1990), l'auteur passe du problème (1.23)

$$\begin{cases} \text{Minimiser } f(x) \\ \text{sous les conditions } g(x) \leq 0, x \geq 0 \end{cases} \quad (1.23)$$

au problème (1.24).

$$\begin{cases} \text{Minimiser } c(i, k)^T y \\ \text{sous les conditions } M^i y + q^i \geq 0, y \geq 0 \end{cases} \quad (1.24)$$

Les indices i et k représentent des pas d'itérations imbriqués. Les quantités c , M et q qui sont utilisées sont liées aux conditions de Kuhn et Tucker.

1.2.2.5 La programmation linéaire par les graphes

Un des ouvrages de référence portant sur les méthodes et les applications de la programmation linéaire par les graphes est (Gondran, 1985).

De nombreuses références issues de l'ENGEES et du Cemagref permettent de comprendre le détail de cette méthode dans le cas des réseaux d'eau potable. Sans être exhaustif, nous pouvons citer (Gilbert, 1988), (Guinaudeau, 1989), (Brémond, 1992), (Ouassi, 1993) et (Gilbert, 1994). Nous présenterons l'application de la programmation linéaire par les graphes aux réseaux d'eau potable au paragraphe 2.1.

Des auteurs se sont intéressés à ces méthodes pour modéliser des phénomènes de transport autres que les réseaux d'eau potable. Nous ne les présenterons pas ici mais on pourra se référer à (Guhl, 1995) pour retrouver ceux traitant notamment des transports ferroviaires ou des réseaux informatiques.

En ce qui concerne les réseaux d'eau potable, d'autres auteurs ont choisi cette méthode pour l'optimisation du fonctionnement d'un réseau d'eau potable. Nous pouvons présenter trois références de George Kuczera (University of Newcastle, NSW, Australia) qui met en place une optimisation d'un réseau sur plusieurs années avec un pas de temps du mois.

Dans (Kuczera, 1989), il développe tous les points qui peuvent être traités en utilisant des fonctions coûts artificiels : la diminution des débordements, l'atteinte d'objectifs de remplissage, la prise en compte de multipompes...

Dans (Kuczera, 1992), il présente le logiciel WATHNET qui gère l'ensemble de la ressource et de l'utilisation de l'eau sur un secteur géographique. Le principe de la modélisation hydraulique de ce logiciel repose sur l'utilisation de la programmation linéaire par les graphes avec des arcs portant des coûts de pénalité pour hiérarchiser les objectifs entre eux.

Enfin, dans (Kuczera, 1993), il compare deux programmes informatiques résolvant l'optimisation par la programmation linéaire par les graphes. Le premier des deux raisonne sur le problème primal alors que le second aborde le problème dual, ce dernier étant le plus performant.

Dans (Guillon, 1991), les auteurs présentent une méthode de gestion de réseaux AEP par un outil appelé SAPHIR. La méthode de résolution est celle des graphes avec une fonction coût comportant une partie linéaire et une partie quadratique (qui exprime les objectifs de niveaux : il faut minimiser le carré de l'écart entre l'objectif et les niveaux calculés). La résolution se fait par la méthode des arcs non conformes (out of kilter).

1.2.2.6 Les méthodes probabilistes

La plupart des méthodes exposées dans les paragraphes précédents sont aujourd'hui enrichies par des aspects probabilistes. C'est pourquoi, nous avons choisi de les regrouper dans un même paragraphe.

Dans (Hillier, 1980), les auteurs tiennent compte de probabilités dans un problème d'organisation de projet du type PERT (Program Evaluation and Review Technique). L'introduction de probabilités leur permet de compléter le temps de réalisation du projet d'un temps minimal (situation la plus optimiste) et d'un temps maximal (situation la plus pessimiste). La résolution se fait par l'utilisation de graphes probabilisés.

Dans (Murat, 1997), l'auteur présente les problèmes d'optimisation combinatoire probabilistes dans les graphes. Elle considère un graphe $G = (V, E)$ et distingue deux cas :

- Si les phénomènes aléatoires portent sur V (les sommets) on parle de problèmes d'optimisation combinatoire probabilistes (POCP) ;
- S'il s'agit des arcs (l'ensemble E) on parle de graphes aléatoires.

Ce deuxième type est assez bien connu. C'est le premier qui est plus nouveau et qui constitue l'axe le plus important de la thèse de Murat.

Toutefois, l'auteur développe d'abord les graphes aléatoires en reprenant les travaux de B. Bollobas. Dans cette théorie, les graphes aléatoires évoluent au cours du temps. Au début, il s'agit d'un ensemble de sommets isolés puis apparaissent des arêtes de façon aléatoire. Il existe une technique permettant de transformer les arêtes d'un graphe en les sommets d'un autre et inversement. Malheureusement, cette technique (appelée line-graph) ne conserve pas les propriétés du graphe initial. Il n'est donc pas possible

d'utiliser cette technique pour résoudre les problèmes de graphes dont les probabilités portent sur les sommets en utilisant des graphes probabilistes.

Dans les POCP, les probabilités traduisent la présence des sommets dans le graphe. L'exemple le plus parlant de problèmes pouvant être modélisés est celui du voyageur de commerce ne devant pas forcément passer dans toutes les villes. L'idée est de ne pas ré-optimiser à chaque fois que l'on connaît la structure du graphe mais de faire un calcul a priori qui pourra être adapté au fur et à mesure de l'évolution du graphe. Ceci est fait pour améliorer la rapidité des calculs. Le problème devient alors davantage de trouver une stratégie de modification de la solution initiale (obtenue dans l'optimisation a priori) vers une nouvelle solution.

Enfin de nombreux travaux ont été consacrés à la prise en compte d'aspects probabilistes dans des programmes linéaires. On peut, en particulier, citer les travaux de (Mészáros, 1996) et de (Eremin, 1984). Dans ce dernier, les auteurs présentent le cadre général d'un programme s'écrivant comme (1.25).

$$\begin{cases} \text{Min } z = cx \\ Ax = b \end{cases} \quad (1.25)$$

Dans (1.25), A , b et c sont constitués de variables aléatoires.

Dans (Aleksandrov, 1984), les auteurs traitent le cas de la gestion de la ressource en eau sur une région. L'aspect probabiliste est inhérent à l'incertitude concernant à la fois les conditions hydrologiques et les conditions de demande (agricole et humaine). La résolution du programme linéaire stochastique obtenu en tenant compte de ces incertitudes est obtenu par passage à un programme convexe déterministe.

Un des chapitres de (Golstein, 1973) est consacré à une vue d'ensemble des programmes linéaires stochastiques. Les auteurs distinguent deux grands types de problèmes : situations comportant un risque qui se traduisent par la possibilité de lier une probabilité au phénomène et situations indéterminées où l'on ne sait pas faire correspondre une loi statistique au problème traité.

Ils envisagent dans un premier temps ce qu'ils appellent une "formulation serrée" qui est utile lorsque l'on cherche un vecteur déterministe comme solution et que les conséquences d'une violation d'une contrainte sont très fâcheuses. Cette formulation revient à prendre le cas le plus défavorable. Le problème peut alors s'écrire comme (1.26).

$$A(q)X \leq B(q), \forall q \in Q, X \geq 0 \quad (1.26)$$

On conviendra d'appeler q le paramètre aléatoire ou la collection de paramètres aléatoires déterminant la réalisation des éléments aléatoires des conditions du problème et Q l'ensemble des valeurs de q à probabilité positive.

Une autre façon de voir le problème consiste à ne pas empêcher l'exclusion de certaines valeurs peu probables dans un premier temps puis d'effectuer une deuxième étape en mettant des pénalités sur une variable d'écart. Cette formulation portera le nom de problèmes à deux étapes.

Une autre définition est fournie par les problèmes à contraintes probabilistes où les écritures habituelles des contraintes sont remplacées par l'équation (1.27).

$$P\left(\sum_{j=1}^n a_{ij}x_j \leq b_i\right) \geq p_i, 0 < p_i \leq 1, \forall i \in \langle 1; m \rangle \quad (1.27)$$

Ceci signifie que la solution d'un tel programme linéaire sera un vecteur qui vérifie la $i^{\text{ème}}$ inégalité avec une probabilité au moins égale à p_i .

Enfin, plusieurs écritures du problème posent la solution X comme étant un vecteur de variables aléatoires. Les auteurs précisent que dans les problèmes où seul le vecteur des contraintes est aléatoire, il est utile d'associer l'inconnue X à la réalisation aléatoire de B par une relation linéaire de la forme : $X = DB$. La matrice D devient alors une inconnue du problème.

Un paragraphe est consacré aux problèmes à contraintes probabilistes. Les auteurs prennent l'exemple de la production de volumes x_j en n périodes ($j = 1, \dots, n$) de façon à minimiser l'espérance mathématique du coût total de production et de stockage du produit, tout en respectant les deux conditions suivantes :

- satisfaire avec une probabilité au moins égale à $p_j^{(1)}$ la demande b_j à distribution normale aléatoire,
- au début de la $j^{\text{ème}}$ période, le stock de produit y_j ne doit sortir des limites $y_{\min}$ et $y_{\max}$ et ce avec une probabilité au moins égale à $p_j^{(2)}$

L'ensemble du programme peut alors s'écrire comme (1.28).

$$\begin{cases} \forall j \in \langle 1; n \rangle, P(x_j + y_j - b_j \geq 0) \geq p_j^{(1)} \\ \forall j \in \langle 1; n \rangle, P(y_{\min} \leq x_j + y_j - b_j \leq y_{\max}) \geq p_j^{(2)} \\ \forall j \in \langle 1; n \rangle, x_j \geq 0, y_j \geq 0 \end{cases} \quad (1.28)$$

Ce problème peut se ramener à un problème de programmation déterministe convexe. Nous verrons plus loin dans ce document une méthode proposée également dans (Golstein, 1973) qui permet de traiter les problèmes probabilistes sans les transformer en problèmes convexes.

1.2.2.7 Les algorithmes génétiques

On peut emprunter à (Ritzel, 1994) la définition du fonctionnement de ces algorithmes. Cet article présente un problème de propagation de polluants. Les auteurs donnent la définition suivante :

Il s'agit d'algorithmes évolutionnaires qui travaillent sur une population. La population est un ensemble d'individus et chaque individu est un point de l'espace de recherche que l'on explore. Le principe est qu'il existe des opérateurs qui manipulent les individus pour faire évoluer la population courante. Cette évolution est obtenue par application aléatoire et itérative de ces opérateurs. Elle consiste à améliorer la performance moyenne d'une population. Les opérateurs sont au nombre de trois :

La sélection qui assure la survie des meilleurs individus

Le croisement et la mutation qui permettent la création de nouveaux individus.

Les algorithmes génétiques sont apparus récemment dans l'optimisation des réseaux d'eau potable. Cette technique est, en particulier, utilisée par l'université d'Exeter en Grande Bretagne.

Dans (Savic, 1995) et (Savic, 1996), les auteurs font le tour des problèmes de gestion et de conception de réseaux d'eau potable qui peuvent être résolus par les algorithmes génétiques :

- tracé d'un réseau ramifié,
- régulation de la pression,
- fonctionnement optimal des pompes,
- calage de réseaux,
- choix optimal d'une extension.

Ces présentations sont précédées d'un rappel sur les algorithmes génétiques. Il apparaît que leur utilisation est intéressante lorsque la seule méthode utilisable est la programmation dynamique.

Dans (Savic, 1997), les auteurs précisent qu'au-delà des coûts de pompage qui représentent l'objectif le plus important, ils choisissent aussi de diminuer l'usure des équipements. Ceci peut se faire en minimisant le nombre d'arrêts/démarrages des pompes. En introduction de cet article, les auteurs rappellent que le Water Research Centre estime que l'on peut économiser 10 % en moyenne des coûts de pompage actuels en mettant en oeuvre une optimisation.

Dans (Mackle, 1995), les auteurs justifient leur choix des algorithmes génétiques comme technique d'optimisation. Ils commencent par citer les différentes méthodes d'op-

timisation utilisées dans ce type de modèles : programmation linéaire, non-linéaire, dynamique ou par les graphes. Ils les considèrent comme étant trop peu transposables (pour chaque réseau il faudrait choisir entre toutes ces méthodes la plus adéquate) et incapables de résoudre de grands problèmes (pas plus de 2 réservoirs ou même 1 réservoir avec plusieurs possibilités de choix de pompes). La fonction coût qu'ils utilisent est principalement constituée des coûts de pompage auxquels on ajoute des coûts d'arrêt/démarrage de pompes ainsi que des fonctions de pénalité pour dépassement des contraintes. De ce fait la fonction objectif est souvent non-linéaire et non-différentiable.

Un exemple avec un réservoir et 4 pompes différentes pouvant alimenter ce réservoir est traité. La méthode ne permet pas d'inclure des contraintes. La fonction objectif est donc enrichie de plusieurs pénalités : manque d'eau dans un réservoir entraînant une insatisfaction de la demande, débordement de réservoir, non-atteinte du niveau objectif final des réservoirs.

1.2.2.8 La décomposition - coordination

La faible vitesse de convergence de certaines méthodes d'optimisation a amené des auteurs à coupler la méthode de programmation dynamique à une méthode de décomposition - coordination. C'est cette méthode qui a été mise en oeuvre par l'équipe de Cohen et Carpentier depuis le début des années 80. Leurs travaux sont relatés dans (Carpentier, 1983) et (Carpentier, 1993).

(Carpentier, 1983) est composée de trois parties. La première partie est consacrée à la théorie de décomposition-coordination développée dans les travaux de Guy COHEN.

La deuxième partie détaille le problème de l'agrégation. Celle-ci est rendue nécessaire par la taille importante en nombre de variables d'état et de décision des sous-réseaux obtenus par un raisonnement hydraulique. Les méthodes d'agrégation reposent principalement sur des similitudes de fonctionnement entre deux réservoirs voisins.

La dernière partie est consacrée à l'application pratique sur le réseau de la Région Parisienne Ouest situé sur les départements des Yvelines et des Hauts-de-Seine. Les résultats sont assez probants surtout si on les compare à ceux obtenus habituellement par la gestion menée par les dispatchers. Toutefois, les comparaisons entre les différentes méthodes d'agrégation laissent apparaître de grandes différences dans les trajectoires obtenues. Ceci est dû au fait que sur un cas réel il est difficile de séparer les différents réservoirs d'un même sous-réseau en deux ou trois classes homogènes et correctement discriminées.

Dans (Carpentier, 1993), la coordination est opérée par la méthode des prix margi-

naux en définissant un sous-réseau comme réseau principal. La décomposition qui est nécessaire dans cette méthode doit aboutir à des sous-réseaux de très petites tailles (au maximum trois réservoirs et une pompe). C'est sur ce point que la méthode a tendance à se compliquer. Ainsi, les auteurs montrent comment il faut agréger certains réservoirs dans les sous-réseaux sur l'exemple d'une partie du réseau de la Région Parisienne Ouest.

En fait, les méthodes mises au point peuvent être utilisées à deux fins différentes : l'optimisation du réseau hors ligne et l'implémentation de méthodes en ligne.

Pour une optimisation hors ligne les étapes suivantes doivent être réalisées :

1. décomposition du réseau en petites unités en choisissant à ce stade les variables de coordination qui seront utilisées dans la suite.
2. élaboration d'un modèle simplifié pour chaque sous-réseau.
3. les deux premiers points étant réalisés une fois pour toutes, il s'agit de faire chaque jour les points suivants :
 - prévisions de consommation pour chaque sous-réseau
 - calcul des trajectoires optimales par la coordination précédée de l'optimisation de chaque sous-réseau par la programmation dynamique discrète
 - désagrégation de chaque sous-réseau pour obtenir les trajectoires réelles.

Le temps de calcul est passé de deux heures en 1985 (sur un Microvax II) à 10 mn aujourd'hui (sur une station Sun Sparc 4/75).

L'application pratique qui a été menée fait apparaître des gains potentiels sur le fonctionnement d'un grand réseau compris entre 3 et 5 %. La faiblesse de ces gains s'explique par deux raisons : la qualité de l'optimisation humaine réalisée jusqu'alors par les dispatchers et le fait que certains sous-réseaux sont très limités au niveau optimisation en raison de contraintes fortes . Il existe, en effet, des sous-réseaux où la taille des réservoirs et des stations de pompage contraint les gestionnaires de réseaux à utiliser les pompes la majeure partie de la journée, réduisant alors la possibilité d'optimisation.

Nous avons utilisé des références traitant de domaines différents de la gestion de réseaux d'eau potable. Ainsi dans (Renaud, 1993) sont présentés les outils dont se servent les gestionnaires de réseaux d'électricité en France. Il n'y a pas de stockage possible mais, par contre, le choix entre les ressources est assez intéressant puisque l'on se trouve dans la situation où certaines ressources peuvent être arrêtées et démarrées facilement alors que d'autres comme les centrales nucléaires sont beaucoup plus inertes.

Les méthodes de résolution utilisées par l'auteur s'appuient sur la programmation dynamique. Le problème est simplifié en utilisant une décomposition des prix et ré-

solu grâce à une méthode de lagrangien augmenté. Le temps de calcul du programme APOGEE réalisé à partir de ces méthodes est de 1 mn sur un IBM 3090.

Dans toutes les références que nous avons vues dans ce paragraphe, chaque sous-système décomposé est ensuite optimisé par la programmation dynamique discrète. Certains auteurs se sont penchés sur la possibilité d'optimiser chaque sous-réseau par des méthodes différentes pour pouvoir améliorer la rapidité de résolution. Ainsi, dans (Zessler, 1989), chaque sous-système est optimisé par la programmation dynamique ou la programmation linéaire. La convergence de la phase de coordination est souvent plus difficile dans ce cas comme cela est montré dans (Gilbert, 1990).

Dans (Ramirez-Beltran, 1993), il s'agit d'optimiser le fonctionnement d'un réseau électrique. Ce réseau est décomposé en sous-réseaux et chaque sous-réseau est optimisé par la programmation linéaire.

1.2.2.9 Les méthodes neuronales

Dans ces méthodes on se passe d'une modélisation hydraulique des phénomènes rencontrés dans la gestion optimale d'un réseau d'eau potable. Dans (Chatenet, 1997), l'auteur met en oeuvre des algorithmes d'apprentissage par renforcement de la politique optimale d'un processus stochastique, appelé Q_learning avec une application aux réseaux d'eau potable. Il s'agit d'apprendre les réactions d'un réseau d'eau à différentes actions possibles en fonction d'un environnement connu et d'en proposer des stratégies de commande qui conduisent à des coûts inférieurs.

1.3 Conclusion

L'examen des deux grandes méthodes de prévision de la demande que nous avons rencontrées nous a permis de montrer les principes généraux de leur mise en oeuvre. A ce stade il est difficile de conclure quant au choix qui pourrait être fait entre les séries temporelles et les réseaux de neurones. Nous serons amené par la suite, comme l'ont fait certains auteurs cités précédemment à comparer ces deux modèles sur les données dont nous disposons. De la même façon, le choix des variables que nous utiliserons en entrée devra être dicté par le type de données que nous chercherons à prédire ainsi que par les conditions d'utilisation future des modèles que nous construisons.

De l'examen des outils disponibles, il ressort qu'il est difficile de concilier une modélisation proche de la réalité avec des temps de calcul permettant la gestion en temps réel des réseaux. Dans le chapitre suivant nous revenons plus en détail sur la façon

dont nous utilisons la programmation par les graphes et la programmation linéaire pour l'optimisation des commandes sur 24 heures. Nous faisons donc le choix de la rapidité des calculs au détriment des phénomènes non-linéaires que certains auteurs ont jugé incontournables. Nous nuancerons ce point de vue plus tard lorsque nous aborderons le problème de l'influence de l'état du réseau sur les transits entre ouvrages.

Chapitre 2

Modèles d'optimisation d'un réseau d'eau potable

2.1 Modélisation par les graphes

2.1.1 Définition générale

Le type de graphe que nous utilisons pour optimiser un réseau AEP est ce que l'on peut appeler un graphe "statique". Ceci signifie que les différentes fonctions attachées au graphe sont des constantes par rapport au temps. Ces différentes fonctions sont principalement les capacités et les coûts de transit entre ouvrages. Le fait que ces fonctions soient constantes est dû à une discrétisation temporelle du graphe.

Dans les faits nous travaillerons souvent sur un horizon d'optimisation de 24 heures avec un pas de discrétisation de l'heure. Par exemple, la modélisation d'un seul réservoir nécessite 24 sommets représentant chacun une heure comme nous pouvons le voir figure 2.1.

2.1.2 Construction du graphe

Pour montrer comment l'on passe d'un réseau AEP au modèle d'optimisation par les graphes prenons l'exemple du réseau représenté figure 2.2.

La modélisation de ce réseau par les graphes peut être représentée figure 2.3.

Sur le graphe de la figure 2.3 on peut distinguer quatre types d'arc différents :

1. Les arcs de remplissage initial partant tous du sommet E

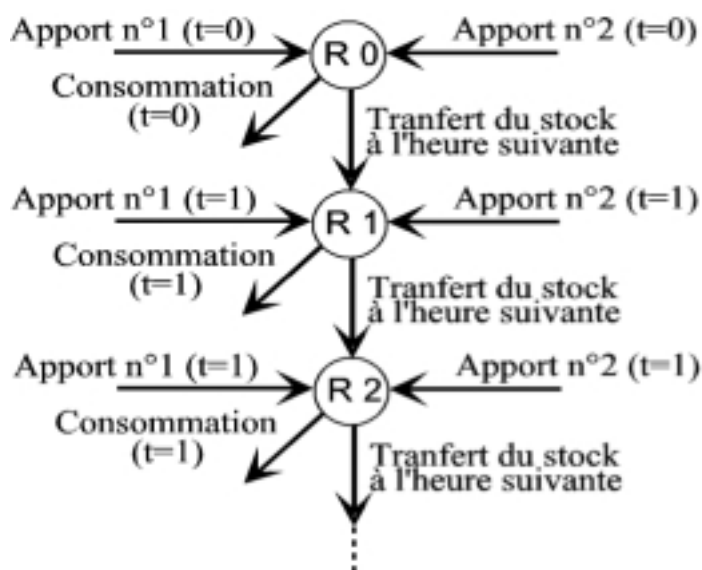


Figure 2.1 : Exemple de décomposition d'un réservoir en fonction du temps

2. les arcs de stock permettant le transfert d'une heure à l'heure suivante sur les deux organes de stockage que sont la bache et le réservoir.
3. Les arcs de consommation qui aboutissent tous au sommet de sortie et qui permettent la satisfaction de la demande sur l'ensemble de l'horizon d'optimisation.
4. Les arcs de transit entre ouvrages que l'on peut subdiviser en trois familles : rivière-bâche, bache-réservoir et forage-réservoir.

Sur chacun de ces arcs figurent une capacité et un coût de transit que l'on peut synthétiser dans le tableau 2.1.

Type d'arcs	Capacité	Coût
1	Stock initial pour les réservoirs et capacité journalière de production pour les sommets représentant une ressource	0
2	Volume utile de l'ouvrage	0
3	Valeur horaire de la consommation prévue	0
4	Capacité hydraulique de l'organe entre les ouvrages (pompes, tuyau,...)	0 pour les transits gravitaires et un coût non nul pour les transits par pompage

Tableau 2.1 : Caractéristiques des arcs pour le graphe exemple

Il est important de noter qu'il est aussi possible d'ajouter un paramètre "capacité

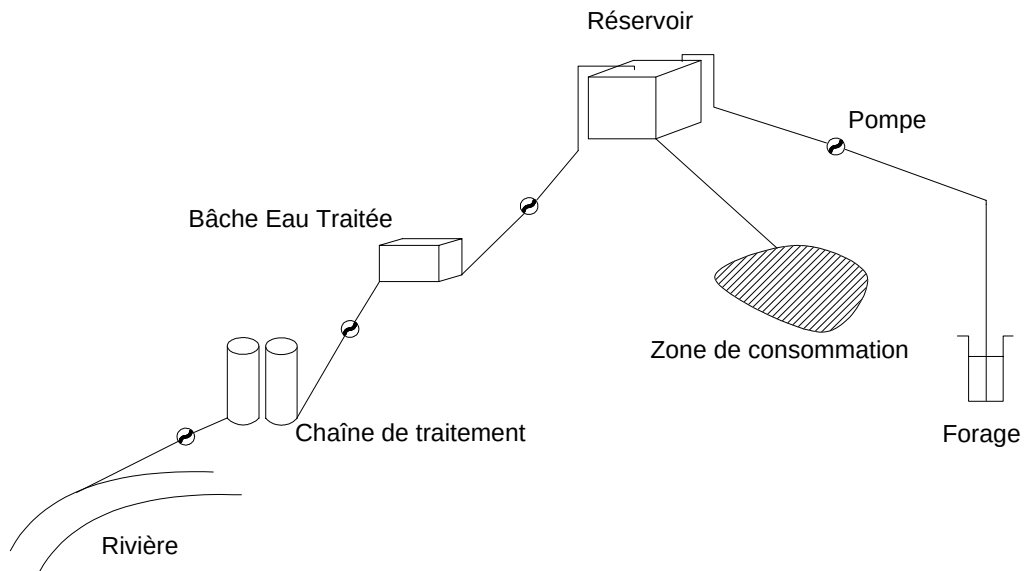


Figure 2.2 : Réseau exemple pour le traitement de la contrainte de continuité

minimale” sur chacun des arcs.

2.1.3 Résolution du modèle

L'algorithme de résolution s'appelle "flot maximal à coût minimal". Le résultat du calcul est alors un flot sur les arcs. Ce flot satisfaisait toutes les demandes en respectant les objectifs de remplissage donnés à chacun des ouvrages de stockage et tout cela au coût minimal. Sur chaque arc, le flot est inférieur ou égal à la capacité de l'arc et, le cas échéant, supérieur à la capacité minimale. Lorsque le flot est égal à la capacité on dit que l'arc est saturé.

Les flots obtenus sont traduits en commande par la fraction de temps de fonctionnement. Prenons l'exemple d'un arc représentant le transit horaire dans une pompe. Si le flot obtenu est égal à 50 % de la capacité de l'arc, la commande consiste à faire fonctionner la pompe pendant 50 % du pas de temps.

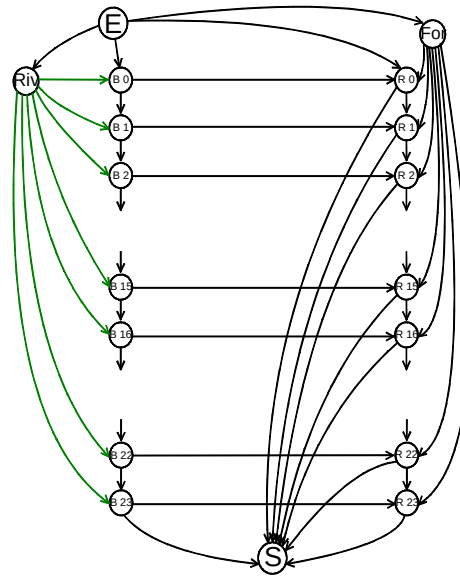


Figure 2.3 : Graphe du réseau exemple

2.2 Modélisation par la programmation linéaire

2.2.1 Horizon et pas de temps

Nous travaillons sur un horizon de 24 heures avec un pas de temps de l'heure. Le choix de l'horizon d'optimisation est dicté par l'objectif, partagé par un grand nombre de gestionnaires de réseaux, de remplir l'ensemble des réservoirs à 6 heures. Cela correspond à la fin des tarifs de nuit en électricité. Le choix de l'heure comme pas de temps est plutôt dicté par les données de consommation que nous utilisons qui sont souvent disponibles à ce pas de temps. Il restera néanmoins possible, pour diminuer les temps de calcul, d'utiliser un pas de temps variable sur la journée (de 1 à 3 heures par exemple). Dans la présentation qui suit et pour ne pas perdre en généralité nous noterons T le nombre de pas de temps.

2.2.2 Vecteurs de commande

2.2.2.1 Définition

Nous appellerons commande la mise en route d'une pompe ou l'ouverture d'une vanne. Soit k le nombre d'organes présents sur le réseau que nous optimisons. On appelle commande élémentaire, un vecteur à k composantes traduisant l'ensemble des com-

mandes du réseau. Il y a alors au plus $I = 2^k$ vecteurs de commande élémentaire comportant chacun k composantes valant 0 ou 1. Nous précisons qu'il y en a au plus 2^k car certains vecteurs de commande potentiels ne sont pas réalisables d'un point de vue physique. C'est le cas, par exemple, de toute commande consistant à mettre en route une pompe débitant dans des canalisations où toutes les vannes seraient fermées.

Les différents vecteurs de commande seront notés par la suite $\vec{K}_i$. A titre d'exemple nous pouvons écrire les vecteurs de commande élémentaire d'un réseau très simple comportant une pompe et deux vannes :

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c} \vec{K}_1 & \begin{array}{c} 0 \\ 0 \\ 0 \end{array} & \vec{K}_2 & \begin{array}{c} 0 \\ 1 \\ 0 \end{array} & \vec{K}_3 & \begin{array}{c} 0 \\ 1 \\ 1 \end{array} & \vec{K}_4 & \begin{array}{c} 0 \\ 0 \\ 1 \end{array} & \vec{K}_5 & \begin{array}{c} 1 \\ 0 \\ 0 \end{array} & \vec{K}_6 & \begin{array}{c} 1 \\ 1 \\ 0 \end{array} & \vec{K}_7 & \begin{array}{c} 1 \\ 1 \\ 1 \end{array} & \vec{K}_8 & \begin{array}{c} 1 \\ 0 \\ 1 \end{array} \end{array}$$

En fonction de la configuration hydraulique de ce réseau certains des vecteurs seront à écarter. Nous conviendrons de noter systématiquement $\vec{K}_1$ le vecteur où toutes les commandes sont à 0. Nous l'appellerons vecteur de référence.

Nous introduisons des variables $\alpha_i([\theta - 1; \theta])$ que l'on notera $\alpha_i(\theta)$ qui représentent la fraction de temps d'application de chacun des vecteurs de commande $\vec{K}_i$ pour chaque pas de temps $[\theta - 1; \theta]$. De cette définition, découle la propriété (2.1).

$$0 \leq \alpha_i(\theta) \leq 1 \quad (2.1)$$

On peut alors exprimer la commande globale sur le réseau par la relation (2.2).

$$\vec{K}(\theta) = \sum_{i=1}^I \alpha_i(\theta) \vec{K}_i \quad (2.2)$$

Le fait d'exprimer des $\alpha_i(\theta)$ revient à discrétiser le temps. On pourra à tout moment revenir à une présentation continue du temps en utilisant la relation (2.3).

$$\alpha_i(\theta) = \int_{\theta-1}^{\theta} \alpha_i(t) dt \quad (2.3)$$

Les fonctions $\alpha_i(t)$ (ou les valeurs des $\alpha_i(\theta)$) représentent donc les inconnues de notre problème d'optimisation.

On peut noter qu'en prenant en compte l'ensemble des vecteurs de commande y compris le vecteur de référence $\vec{K}_1$ nous aurons la relation (2.4).

$$\forall \theta \in \langle 1; T \rangle, \sum_{i=1}^I \alpha_i(\theta) = 1 \quad (2.4)$$

2.2.2.2 Réduction de l'espace des vecteurs de commande

On peut, en fait, classer les différentes commandes élémentaires par groupes définis de la façon suivante : “l'effet d'une commande élémentaire d'un groupe est indépendant des états des commandes élémentaires des autres groupes”. Prenons l'exemple de la figure 2.4.

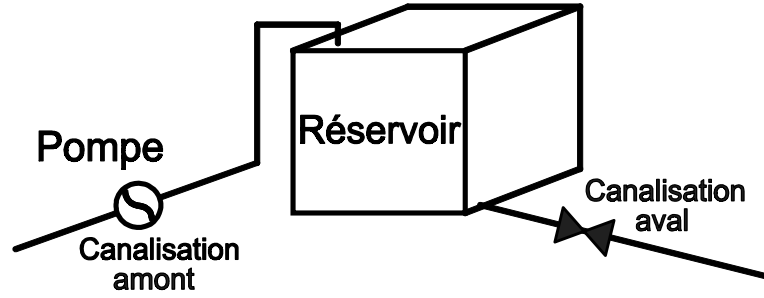


Figure 2.4 : Séparation des commandes par groupes

On voit sur ce schéma que la mise en marche de la pompe n'aura pas d'effet sur le débit circulant dans la canalisation aval. De la même façon, l'ouverture ou la fermeture de la vanne ne modifiera en rien le débit transitant dans la canalisation amont lorsque la pompe se mettra en route.

Soit G le nombre de groupes de commandes élémentaires ainsi définis. Chaque groupe g est caractérisé par les vecteurs de commande $\vec{K}_i^g$ construits avec les I_g commandes élémentaires de ce groupe (avec $i \in [1, I_g]$ et $I_g = 2^{k_g}$) et appliqués pendant $\alpha_i^g(\theta)$ durant le temps θ .

L'intérêt de ce découpage est de réduire le nombre de variables α puisque nous avons la relation $\sum_{g=1}^G 2^{k_g} \ll 2^k$ avec $\sum_{g=1}^G k_g = k$.

On peut écrire, pour chaque groupe, la même contrainte que celle de l'équation (2.4).

$$\forall g \in \langle 1; G \rangle, \forall \theta \in \langle 1; T \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \quad (2.5)$$

2.2.3 Débits entrant dans les réservoirs

Le résultat de la commande définie au paragraphe précédent se traduira par des débits transitant sur le réseau. Pour les caractériser nous choisirons de retenir les débits entrant dans chaque réservoir que nous noterons Q_{ir} avec i indice caractérisant la commande

appliquée et r le réservoir. Nous aurons évidemment là aussi :

$$\forall r \in \langle 1; R \rangle, Q_r(\theta) = \sum_{i=1}^I \alpha_i(\theta) Q_{ir}(\theta) \quad (2.6)$$

Avec $Q_r(\theta)$ qui représente le volume entrant dans le réservoir r durant le pas de temps θ .

Pour tenir compte de la réduction de l'espace des commandes, nous écrirons :

$$\forall r \in \langle 1; R \rangle, Q_r(\theta) = \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) Q_{ir}^g(\theta) \quad (2.7)$$

2.2.4 Les consommations

Nous noterons $C_r(t)$ la demande instantanée à la sortie du réservoir r et $C_r(\theta) = C_r([\theta - 1, \theta]) = \int_{\theta-1}^{\theta} C_r(t) dt$ le volume demandé en sortie du réservoir r au cours du pas de temps $[\theta - 1, \theta]$.

2.2.5 Les contraintes de volume

Pour chaque réservoir les contraintes que nous exprimons traduisent que :

- à chaque pas de temps, le volume courant, $V_r(\theta)$, doit être compris entre le volume maximal, $V_{r \max}$, et minimal, $V_{r \min}$,
- en fin d'horizon d'optimisation, on cherche à atteindre le niveau maximal, $V_{r \max}$.

2.2.6 La fonction économique

Il s'agit de minimiser les coûts occasionnés par les commandes passées sur le réseau. Nous noterons $\pi_i(\theta)$ le coût de la commande i au cours du pas de temps $[\theta - 1, \theta]$. On retrouve alors une forme proche de celle que nous avons déjà utilisée dans les paragraphes précédents.

$$z_{\min} = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^I \alpha_i^g(\theta) \pi_i^g(\theta) \quad (2.8)$$

2.2.7 Ecriture complète du modèle

$$\left\{ \begin{array}{l} z_{\min} = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) \\ \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^{\theta} C_r(t) \leq V_{r \max} \\ \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^{\theta} C_r(t) \geq V_{r \min} \\ \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^T C_r(t) = V_{r \max} \\ \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \\ \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \forall i \in \langle 1; I_g \rangle, \alpha_i^g(\theta) \geq 0 \end{array} \right. \quad (2.9)$$

Il ne s'agit que d'une façon d'écrire notre modèle. Suivant les méthodes que nous mettrons en oeuvre nous pourrions choisir une autre présentation faisant, par exemple, intervenir explicitement les variables de stock.

En séparant les parties constantes et variables nous pouvons présenter ce problème d'optimisation sous la forme matricielle (2.10).

$$\left\{ \begin{array}{l} z_{\min} = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) \\ A\alpha \leq B \end{array} \right. \quad (2.10)$$

Dans cette écriture α représente le vecteur des inconnues $\alpha_i^g(\theta)$, le vecteur B représente le vecteur des seconds membres. Il est constitué de constantes du problème du type

$$\sum_{t=1}^{\theta} (C_r(t) + V_{r \max} - V_r(0))$$

La matrice A traduit à la fois les contraintes hydrauliques du réseau (débits entrants dans les réservoirs en fonction de la commande appliquée) et les contraintes de définition des $\alpha_i^g(\theta)$. L'expression (2.9) montre que la matrice A a une forme régulière. Plus précisément elle est triangulaire par blocs comme nous pouvons le voir sur la figure 2.5.

Dans toute la suite de ce document nous noterons $C_{r\theta} = \sum_{t=1}^{\theta} C_r(t)$.

2.3 Conclusion

La programmation par les graphes ainsi que la programmation linéaire supposent donc le même travail préparatoire à la modélisation. Il est nécessaire dans les deux cas de définir les flux unitaires conséquences de la mise en oeuvre des commandes. La

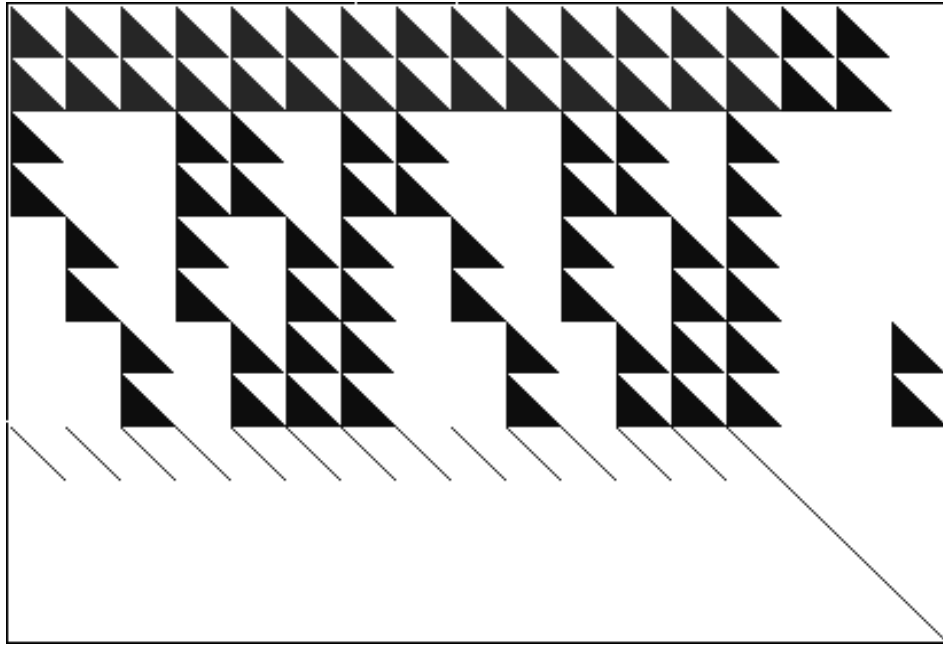


Figure 2.5 : Matrice des contraintes du problème linéaire

modélisation par les graphes semble a priori plus explicite en donnant une interprétation des hypothèses et des résultats plus proche de la réalité. Néanmoins, nous verrons dans la suite de ce document que la programmation linéaire permet de couvrir plus de cas particuliers et ce, notamment, par la possibilité, à condition de changer d'algorithme de résolution, d'astreindre certaines variables à être entières.

Dans le prochain chapitre, nous nous attachons à prévoir les consommations $C_r(t)$.

Chapitre 3

Prévision de la demande

Dans la bibliographie actuelle, deux grands types d'outils sont développés pour la détermination d'une fonction de demande. Il s'agit soit de modèles de séries temporelles soit de modèles à base de réseaux de neurones. Nous nous intéresserons principalement aux modèles à base de réseaux de neurones. Nous présenterons également une comparaison de ces deux grands types de modèles sur nos données et sur des données générées artificiellement.

3.1 Les données

Dans le cadre d'une convention signée entre le Cemagref, l'Enges, la Compagnie générale des Eaux et Anjou Recherche nous pouvons utiliser des données stockées au pas de temps de l'heure depuis l'été 1993 sur le Syndicat Intercommunal des Eaux de la Région de Sarralbe. Ces données représentent la valeur d'index de compteurs situés à la sortie des réservoirs sur l'ensemble du réseau. Ces données couvrent une période qui semble suffisante. Toutefois, il manque des données, parfois sur plusieurs jours consécutifs, du fait de défaillances du système de mesures ou de rapatriement de la mesure. De plus, une difficulté importante réside dans le pré-traitement des données : il est nécessaire de discriminer les informations qui représentent réellement la consommation de la zone desservie par le réservoir de celles traduisant un transit vers d'autres réservoirs. Après plusieurs essais infructueux, une première zone de consommation semble pouvoir être utilisée. Dans cette zone, appelée "zone vallée", persiste néanmoins un certain nombre de valeurs "suspectes". Ces valeurs qui s'écartent significativement des valeurs habituelles sont des dysfonctionnements d'ordre hydraulique (casses ou fuites) ou des erreurs de mesure (capteurs, télétransmission ou archivage des données). Avant de pouvoir uti-

liser les données nous avons donc procédé à un pré-traitement empirique qui a consisté à éliminer les valeurs s'écartant sensiblement de la moyenne.

La zone utilisée représente une zone rurale de faible dimension. L'intérêt de ce type de réseau est la variabilité importante des données de consommation. Nous ne présentons pas de comparaison des zones urbaines mais il est possible de se reporter à (Ferstler, 1998) qui montre des résultats obtenus sur une partie de la Communauté Urbaine de Strasbourg avec les méthodes de réseaux de neurones exposés dans ce chapitre.

Pour la construction de modèles de prédiction de la demande, plusieurs auteurs ont utilisé des données météorologiques ou climatiques. Nous avons choisi de nous affranchir de ce type de données et ceci pour quatre raisons :

- les liens entre ces données météorologiques et la consommation semblent difficiles à établir car il faut choisir le facteur explicatif adéquat (nombre de jours consécutifs de sécheresse, nébulosité faible, seuil de température,...) ;
- L'utilisation de prévisions météorologiques revient à faire de la prédiction de demande à partir de prévision ;
- Les modèles que nous construisons devront pouvoir être implantés sur un site central où la télégestion du réseau est assurée. Ceci suppose que les données d'entrée du modèle devront être facilement accessibles ce qui n'est pas forcément le cas pour des données météorologiques ;
- Les données météorologiques sont implicitement contenues dans les consommations passées.

3.2 Prédvisions à l'heure suivante

Nous avons testé deux types de modèles, l'un à base de réseaux de neurones et le second construit à partir de séries temporelles.

3.2.1 Construction du modèle de réseau de neurones

Dans un premier temps nous construisons un réseau de neurones qui permet de prédire la consommation d'une heure en connaissant les consommations passées.

Il faut alors choisir les variables explicatives à utiliser en entrée. Celles-ci seront de deux types : des variables exogènes et des variables endogènes. Les variables exogènes initialement recensées étaient les suivantes :

- le mois de l’année,
- le jour de la semaine,
- l’heure de la journée,
- le jour considéré est un jour férié,
- le jour considéré correspond à une période de vacances scolaires.

Les variables endogènes correspondent pour leur part aux consommations des heures précédentes.

Après avoir testé plusieurs choix possibles nous avons retenu le réseau de neurones de la figure 3.1.

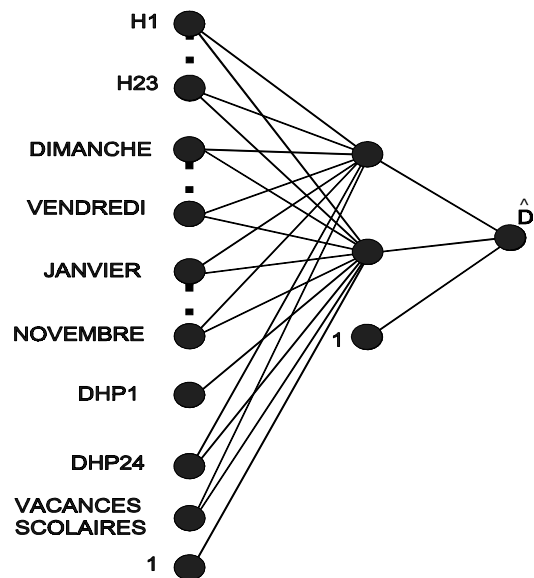


Figure 3.1 : Prédiction à l’heure suivante par réseau de neurones

En plus du biais, ce réseau comporte 65 entrées réparties de la façon suivante :

- 23 neurones prenant la valeur 0 ou 1 pour définir l’heure de la journée correspondante à la sortie que l’on veut prédire avec comme règle que si tous ces neurones valent 0 l’heure vaut 24 ;
- 6 neurones permettant de la même façon de définir le jour de la semaine ;
- 11 neurones définissant le mois de l’année ;
- 24 neurones appelés DHP_i qui prendront la valeur de la demande de l’heure précédente. Concrètement lorsque l’on cherchera à prédire la demande de l’heure i ,

DHP_i vaudra la demande de l'heure pr c dente en m^3 et tous les autres DHP_j avec $j \neq i$ vaudront 0 ;

- 1 neurone valant 0 ou 1 pour pr ciser si l'heure o  l'on effectue la pr vision fait partie d'une journ e de vacances scolaires.

Nous utilisons un logiciel de l'universit  Laval de Qu bec, le logiciel Nnfit, qui nous permet de construire un r seau de neurones   une couche cach e. Ce logiciel permet  galement de tester plusieurs nombres de neurones dans cette couche cach e afin de retenir celui donnant une bonne pr vision tout en limitant le nombre de param tres du mod le. La figure 3.1 pr sente le r seau avec deux neurones, en plus du biais, dans la couche cach e mais nous donnerons aussi les r sultats comparant plusieurs choix possibles.

Pour tester la qualit  de la pr vision nous utilisons une m thode habituelle de s paration- valuation. Une partie des donn es (le fichier d'apprentissage) sert   la construction du mod le c'est- -dire au calcul des poids des diff rentes connexions alors que l'autre partie (le fichier de g n ralisation) sert   valider le mod le obtenu.

Le logiciel Nnfit permet, en outre, lorsqu'un tel d coupage a  t  effectu  de limiter le nombre d'it rations pour  viter le ph nom ne de sur-apprentissage (ou over fitting). Ce ph nom ne se traduit par de tr s bons r sultats sur la partie du fichier qui a servi   l'apprentissage et des r sultats nettement moins bons lorsque le mod le s'applique   d'autres entr es (c'est dans ce cadre qu'est utilis e la partie du fichier servant   la g n ralisation).

Dans notre cas nous utiliserons les donn es de mi-93   fin 1996 pour l'apprentissage (19 200 individus horaires) et les donn es de l'ann e 1997 (5 400 individus) pour valider ce mod le.

3.2.2 R sultats du mod le

Plusieurs indicateurs nous permettent de caract riser la qualit  du mod le. Dans un premier temps, nous pouvons fournir les coefficients de corr lation entre les donn es mesur es et les donn es pr dites. Il y aura en fait deux coefficients de corr lation ; l'un pour le fichier d'apprentissage et l'autre pour le fichier de g n ralisation. Les deux sont indissociables pour les raisons que nous avons  voqu es pr c demment.

Afin de choisir parmi les diff rentes architectures test es, obtenues en faisant varier le nombre de neurones dans la couche cach e, nous pr senterons le nombre de poids de chaque mod le. L'objectif d'un bon mod le est, en effet, de concilier un coefficient de

corrélation élevé avec un nombre de poids le plus faible possible. Le nombre de poids s'obtient par : $(I + 1)J + (J + 1)K$ avec I , nombre de neurones dans la couche d'entrée (biais non compris), J , nombre de neurones dans la couche cachée (biais non compris) et K , nombre de neurones dans la couche de sortie. Le nombre de poids traduit a priori une diminution des degrés de liberté d'un modèle. On peut donc dire qu'un modèle ayant un nombre de poids élevé risque d'entraîner un sur-apprentissage car le calage des poids aura permis d'approcher la fonction présente dans le fichier d'apprentissage même si cette dernière comporte des erreurs systématiques.

Nous comparerons également les distributions (moyenne et écart type) pour les valeurs observées et mesurées. Enfin nous nous intéresserons aux erreurs entre ces deux vecteurs en observant la distribution de ces erreurs et en s'attardant sur les valeurs anormalement fortes.

Commençons donc par examiner la corrélation entre les valeurs mesurées et les valeurs prédites par notre modèle :

Nb de neurones dans la couche cachée	Nb de poids	Fichier d'apprentissage			Fichier de généralisation		
		Coeff de corrélation	Moyenne (réf. 29.38)	Ecart-type (réf. 14.28)	Coeff de corrélation	Moyenne (réf. 33.33)	Ecart-type (réf. 15.26)
2	135	0.969	29.39	13.82	0.949	32.54	14.44
3	202	0.970	29.39	13.84	0.947	32.53	14.52
4	269	0.972	29.38	13.88	0.947	32.53	14.49

Tableau 3.1 : Comparaison du nombre de neurones dans la couche cachée

Le tableau 3.1 montre que l'augmentation du nombre de neurones dans la couche cachée n'augmente pas significativement la qualité des résultats obtenus. Nous retiendrons le modèle avec 2 neurones dans la couche cachée qui allie de bons résultats à un nombre de poids faible.

Nous remarquons également que quelque soit le nombre de neurones dans la couche cachée l'écart type des valeurs prédites est toujours plus faible que celui des valeurs réelles. Le modèle travaille donc sur une plage plus réduite de valeurs que la réalité.

Le dernier critère que nous utilisons est l'examen de la distribution de l'erreur. Commençons par définir cette erreur : $\text{erreur} = (\text{valeur mesurée} - \text{valeur prédite})$. Nous nous intéresserons plus précisément à l'erreur du modèle comportant deux neurones dans la couche cachée.

L'examen de la répartition des erreurs, sans tenir compte de l'aspect temporel, se

présente comme figure 3.2.

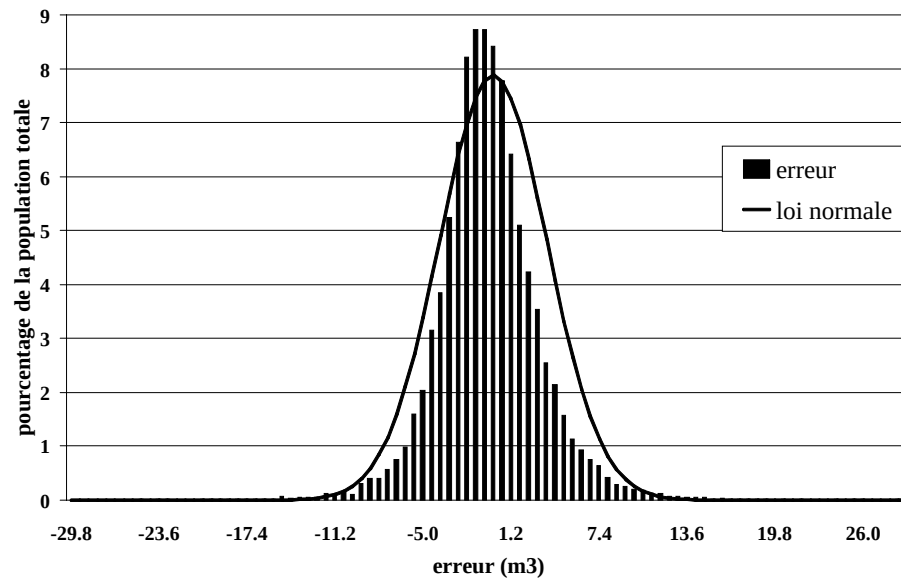


Figure 3.2 : Distribution des erreurs

L'allure de la répartition de la figure 3.2 semble normale. Toutefois un test du Khi-2 invalide cette hypothèse. Pour confirmer ce test nous avons également représenté sur cette même figure la répartition gaussienne théorique avec la moyenne et l'écart-type de la vraie distribution.

Nous avons, par ailleurs, isolé les erreurs particulièrement élevées. Pour ce faire nous avons retenu comme critère le fait que la valeur absolue de l'erreur de prédiction soit supérieure à 15 m^3 , soit environ la valeur de l'écart-type des données. L'examen de ces erreurs nous amène à nous pencher à nouveau sur les données de départ. Nous constatons que les variables caractérisant la demande de l'heure précédente ont un poids très important. Ceci a été confirmé par deux études différentes que nous avons effectuées :

- la première consistant à construire un modèle comportant d'autres demandes des heures passées en plus de celle de l'heure précédente : cela n'améliore pas le modèle et le poids de la variable "demande de l'heure précédente" reste le même.
- la deuxième consistant à l'inverse à construire un modèle où la seule variable d'entrée est la "demande de l'heure précédente". Nous obtenons alors un modèle donnant un coefficient de corrélation entre valeurs à prédire et valeurs prédites d'environ 0,9.

Ce poids important de la demande de l'heure précédente fait que lorsqu'il y a une variation brusque entre deux heures le résultat est souvent mauvais. Cet écart brusque peut être dû à trois choses : soit à une erreur de mesure qui peut se traduire par des demandes en dent de scie sur plusieurs heures, soit à une surconsommation anormale de type casse hydraulique sur le réseau soit, enfin, à une consommation réelle mais significativement plus forte ou plus faible qu'à l'heure précédente. On voit, par cette énumération, qu'il est difficile de distinguer les valeurs à rejeter de celles à conserver.

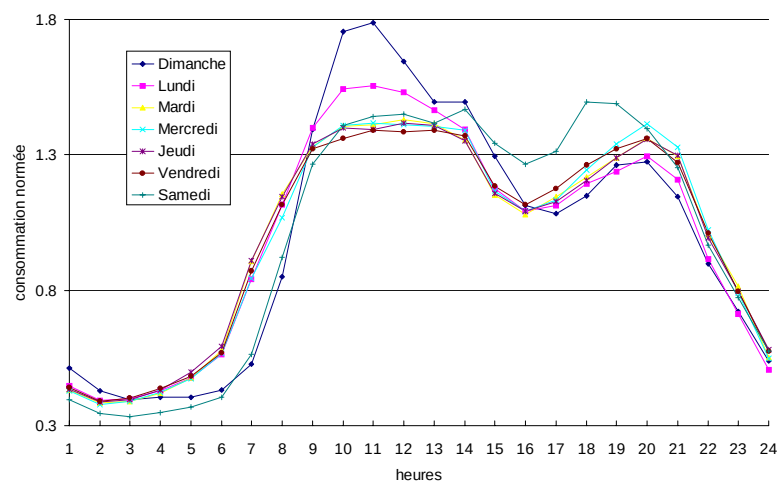


Figure 3.3 : Spectre des consommations horaires

3.2.3 Retour sur le nombre de poids

Nous avons affirmé précédemment que le nombre de poids d'un bon modèle devait être le plus faible possible. Or le nombre de poids du modèle que nous venons de présenter est au minimum de 135. Ceci est dû au grand nombre de neurones en entrée. En regardant plus en détail les entrées que nous utilisons, nous remarquons qu'un grand nombre d'entre elles valent 0 ou 1. Ceci signifie que nous avons réuni dans un seul réseau de neurones un modèle qui aurait pu être représenté par plusieurs réseaux. Il aurait été possible, par exemple, de construire un réseau de neurones par jour de la semaine. Nous aurions ainsi construit 7 réseaux de neurones avec 123 poids chacun dans le cas où il y a deux neurones dans la couche cachée.

Pour montrer que le résultat est bien comparable avec les deux méthodes, nous présentons pour les données à prédire et les données prédites les spectres de consommation par jour de la semaine. Ces spectres représentent la consommation normalisée, obtenue en

divisant, à chaque heure, la consommation par la consommation moyenne de la journée. C'est sur ces données normées, représentées sur les figures 3.3 et 3.4 que nous pouvons le mieux observer la régularité dont nous parlions au début.

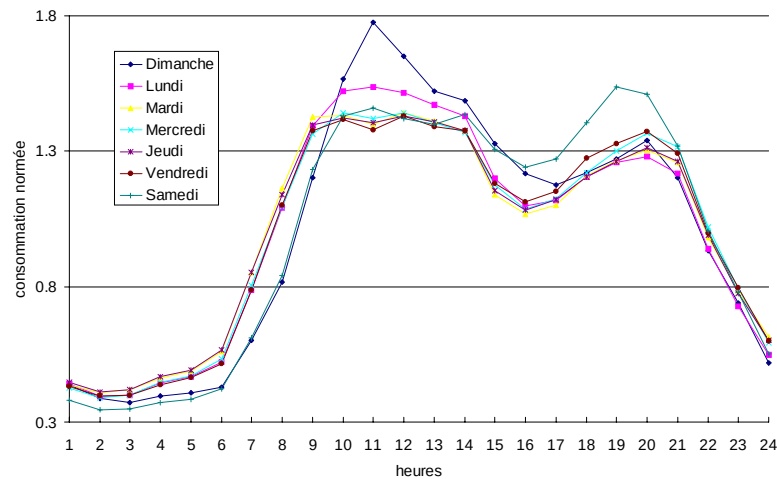


Figure 3.4 : Spectre des prévisions

Nous retrouvons bien les mêmes formes de courbe par jour de la semaine. Le raisonnement que nous avons mené avec les jours de la semaine peut être reproduit avec le mois de l'année ou l'heure de la journée et bien entendu avec les combinaisons de ces facteurs.

3.3 Construction du modèle de séries temporelles

Parallèlement à la construction de modèles par réseaux de neurones nous avons voulu connaître le comportement des séries temporelles avec les mêmes données. Pour ce faire nous avons comparé les modèles prédisant l'heure suivante. Nous avons tout d'abord effectué cette comparaison avec les données de la zone vallée puis nous avons travaillé sur des données générées artificiellement.

3.3.1 Données de la zone "Vallée"

Pour que la comparaison soit intéressante nous avons pris les mêmes entrées. Le logiciel qui a été utilisé pour la construction de ces modèles de séries temporelles est le logiciel SAS.

Le mod le construit peut s' crire sous la forme (3.1).

$$D(t) = \beta_0 + \sum_{i=1}^{24} \beta_i DHP_i + \sum_{j=1}^{23} \beta_{j+24} H_j + \sum_{k=1}^6 \beta_{k+47} JS_k + \sum_{l=1}^{11} \beta_{l+53} M_l + \beta_{65} VAC + \varepsilon_t \quad (3.1)$$

Le calage des diff rents coefficients β est effectu  avec la m me base de 19 200 individus horaires que celle utilis e pour les r seaux de neurones. Les r sultats obtenus sont comparables   ceux du mod le   base de r seaux de neurones. En particulier, l' tude de la distribution des erreurs fait, l  aussi, appara tre un bruit non blanc. Le logiciel SAS permet d'aller plus loin en examinant l'existence d'une corr lation temporelle (ou autocorr lation) entre les erreurs. Celle-ci est effectivement observ e et l'on recale les coefficients de la r gression en tenant compte de cette autocorr lation. On obtient alors le mod le (3.2).

$$D(t) = \delta_0 + \sum_{i=1}^{24} \delta_i DHP_i + \sum_{j=1}^{23} \delta_{j+24} H_j + \sum_{k=1}^6 \delta_{k+47} JS_k + \sum_{l=1}^{11} \delta_{l+53} M_l + \delta_{65} VAC + F_t \quad (3.2)$$

Dans l' quation (3.2) F_t s' crit $F_t = - \sum_{m=1}^{25} \gamma_m F_{t-m} + \varepsilon_t$

Les coefficients δ obtenus sont assez proches des β du premier mod le.

3.3.2 Comparaison des deux m thodes sur les donn es r elles

Le logiciel Nnfit ne permet pas de construire des mod les de r seaux de neurones tenant compte de l'autocorr lation des r sidus. Pour que cela soit possible il faudrait que l'apprentissage puisse se faire   l'aide d'une m thode des moindres carr s g n ralis s. Cette diff rence rend la comparaison des deux m thodes difficile. N anmoins la comparaison des erreurs fortes obtenues pour chacun des mod les conduit   pointer les m mes individus. Le coefficient de corr lation entre les valeurs attendues et les valeurs pr dites reste  galement tr s proche pour les deux mod les.

Nous avons n anmoins voulu  tudier la corr lation des r sidus du mod le par r seaux de neurones. Nous avons, alors, construit une r gression lin aire multiple du type :

$$\varepsilon(t) = \sum_{i=1}^{25} \lambda_i \varepsilon(t-i) \quad (3.3)$$

Nous n'arrivons qu'  expliquer une faible partie de l'erreur et le bruit r siduel n'est toujours pas Gaussien. N anmoins il est int ressant de constater que les quatre r gresseurs qui ont la probabilit  d'explication la plus forte pour le mod le lin aire que nous construisons sont dans l'ordre d'importance $\varepsilon(t-24)$, $\varepsilon(t-23)$, $\varepsilon(t-25)$ et $\varepsilon(t-1)$. Dans

le modèle par séries temporelles nous retrouvons ces mêmes retards avec une probabilité plus élevée. Nous pouvons donc penser que l'ajout d'information sur la consommation à $(t - 24)$ pourrait encore améliorer nos modèles.

3.3.3 Comparaison sur des données fictives

Une autre idée de comparaison des deux modèles consiste à générer des données grâce aux séries temporelles et de tester ces données dans la construction d'un modèle par réseaux de neurones. Le logiciel SAS permet la génération d'un fichier respectant la structure des données réelles. En particulier, on introduit un bruit autocorrélé sur 25 pas de temps en utilisant bien entendu les coefficients d'autocorrélation obtenus lors de la construction des précédents modèles.

Le fichier construit établit la demande horaire pour une période allant du 1^{er} janvier 1990 au 31 décembre 1999.

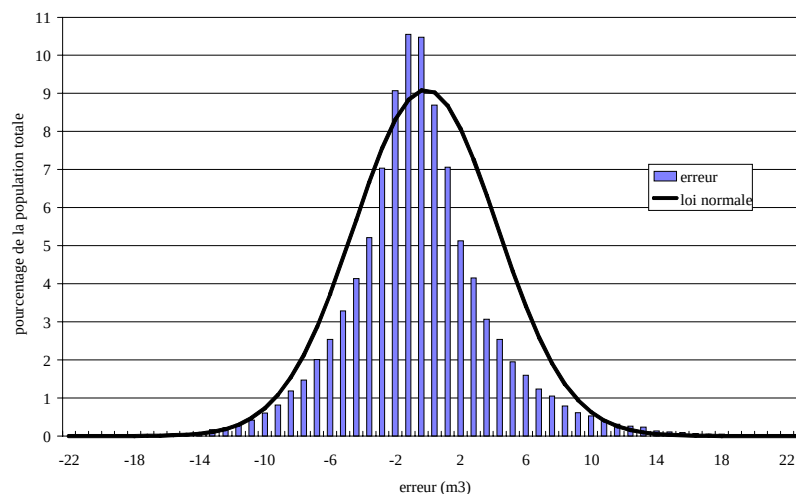


Figure 3.5 : Répartition des erreurs dans la prévision des données artificielles

Pour la construction du modèle de réseaux de neurones nous avons séparé ce fichier en deux : la base d'apprentissage est constituée des demandes horaires du 1^{er} janvier 1990 au 31 décembre 1996 alors que la généralisation est faite sur les données du 1^{er} janvier 1997 au 31 décembre 1999. Nous n'examinons ici que le réseau de neurones avec 2 neurones, en plus du biais, dans la couche cachée.

La répartition des erreurs sur la partie des données ayant servi à l'apprentissage a la forme représentée figure 3.5.

Fichier d'apprentissage				Fichier de généralisation			
Nombre d'individus	Coefficient de corrélation	Moyenne (réf. 29.27)	Ecart-type (réf. 14.53)	Nombre d'individus	Coefficient de corrélation	Moyenne (réf. 29.40)	Ecart-type (réf. 14.68)
61367	0.953	29.27	13.86	26280	0.954	29.38	13.96

Tableau 3.2 : Résultats des prévisions sur les données artificielles

Pour ces données artificielles le fait de ne pas tenir compte de l'autocorrélation des résidus est très handicapant quant à la qualité de la prévision obtenue.

3.4 Prévisions à plus long terme

Nous avons jusqu'ici présenté des modèles de prévision de demande d'heure en heure. Or, nous aurons besoin dans l'optimisation d'un réseau d'eau potable de connaître la demande sur plusieurs heures consécutives. Idéalement les modèles à bâtir devraient prévoir la demande sur 24 heures ce qui correspond à l'horizon d'optimisation que nous utilisons. Dans les faits notre optimisation débutant à 6 heures du matin avec l'ensemble des réservoirs pleins, les conséquences d'erreurs de prévision en début d'horizon sont faibles. Ce n'est que plus tard dans la journée, lorsque le risque de voir des réservoirs se vider sera plus fort, qu'il sera nécessaire d'avoir une prévision fiable. C'est pourquoi nous avons examiné un modèle prévoyant les demandes de 13 heures à 24 heures.

Les faibles écarts constatés entre les modèles de séries temporelles et de réseaux de neurones au paragraphe précédent nous amène à ne plus construire cette comparaison pour la prévision à long terme. Nous nous intéressons uniquement aux modèles de prévision par réseaux de neurones mais il aurait également été possible de construire des modèles à long terme à l'aide des séries temporelles. Nous envisageons deux types de réseau de neurones différents.

3.4.1 Les deux modèles de réseaux de neurones

3.4.1.1 Modèle à une sortie

Le premier modèle est du même type que celui utilisé pour prévoir la demande de l'heure suivante avec des entrées différentes. Il peut se représenter comme figure 3.6.

Les entrées utilisées sont comparables à celles du modèle présenté figure 3.1. Les neurones de l'heure prédite prennent cette fois les valeurs 13 à 23 au lieu de 1 à 23. De plus nous utilisons maintenant 12 neurones donnant la valeur de la consommation des

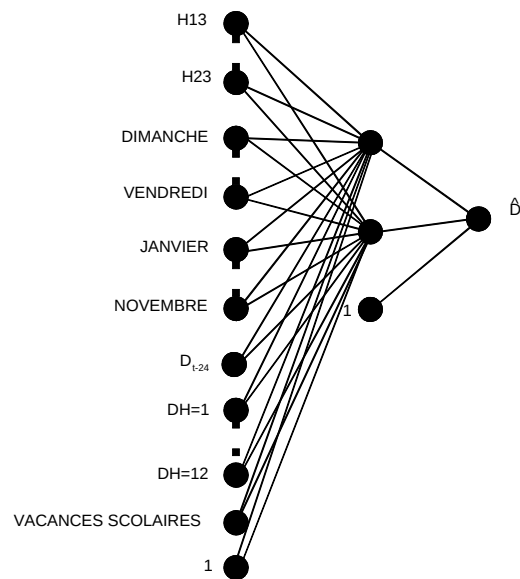


Figure 3.6 : Prédiction à long terme. Réseau de neurones à une sortie

12 premières heures de la journée ($DH = 1$ à $DH = 12$). Enfin, utilisant les résultats du premier modèle nous avons ajouté un neurone donnant la valeur de la demande de la même heure la veille (D_{t-24}).

3.4.1.2 Modèle à 12 sorties

L'autre modèle que nous utilisons est un réseau de neurones à plusieurs sorties. Le modèle est alors plus compact mais l'information en entrée est moins fournie puisqu'il ne peut y avoir une série de neurones donnant l'heure de la prédiction. Le réseau est alors représenté figure 3.7.

3.4.2 Les résultats

Nous utilisons à nouveau les données réelles de 1993 à 1996. Pour constater l'évolution de la qualité de la prédiction au cours du temps nous allons représenter deux critères : le coefficient de corrélation entre les valeurs attendues et les valeurs prédites, figure 3.8 et la moyenne des erreurs figure 3.9.

3.4.2.1 Modèle à une sortie

Dès le début de l'horizon de prédiction, les coefficients de corrélation sont moins bons que ceux obtenus pour le modèle de prédiction à l'heure suivante. Il est intéressant de

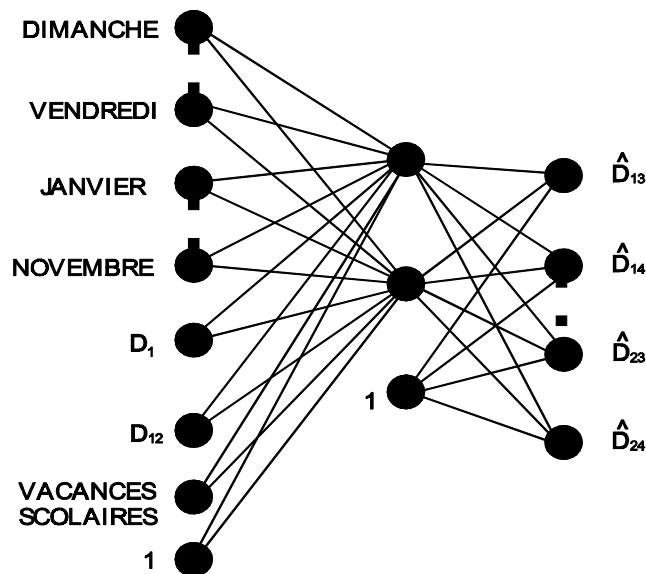


Figure 3.7 : Prédiction à long terme. Réseau de neurones à plusieurs sorties

noter que ces coefficients restent à une valeur correcte jusqu'à 22 heures. Les prévisions de la fin de l'horizon ne sont plus fiables.

Le cumul des erreurs ne semble pas suivre une règle particulière au cours de l'horizon de prévision. Il s'agit bien d'un cumul en valeur algébrique. Les valeurs proche de 0 peuvent donc traduire une compensation des erreurs d'une journée sur l'autre.

3.4.2.2 Modèle à 12 sorties

Les coefficients de corrélations, représentés figure 3.10, sont meilleurs avec ce modèle à 12 sorties qu'avec le modèle précédent et comparables, en début d'horizon, avec ceux obtenus pour le modèle de prévision à l'heure suivante. La dégradation des résultats au cours des deux dernières heures de l'horizon reste vraie pour ce modèle.

Les erreurs constatées, représentées figure 3.11, sont plus faibles avec ce modèle à 12 sorties. Il n'y a toujours pas de règle particulière d'évolution du cumul des erreurs au cours de l'horizon de prévision.

3.5 Conclusion

Malgré l'impossibilité que nous avons rencontré de pouvoir comparer la distribution des erreurs pour chacun des modèles, nous concluons néanmoins à la possibilité de

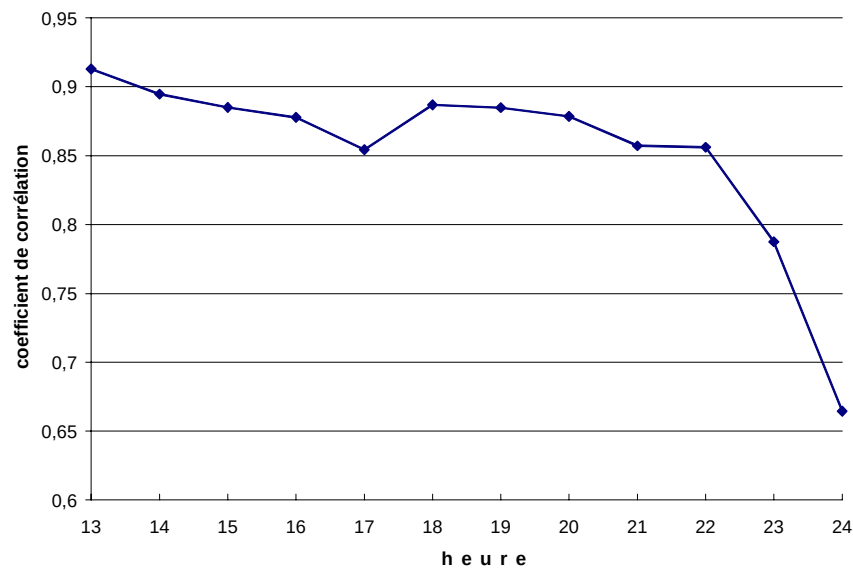


Figure 3.8 : Coefficients de cor lation du mod le   une sortie

pouvoir utiliser l'un ou l'autre des mod les pour la pr vision de la demande   l'heure suivante. Les  carts entre les deux mod les sont notamment tr s faibles en comparaison des erreurs syst matiques qui sont pr sentes   l'origine dans les donn es que nous avons archiv es. La pr vision   plus long terme donne des r sultats acceptables pendant plusieurs heures. La d gradation observ e en cours de journ e nous am nera lors de la mise en oeuvre de ces m thodes sur site   d couper notre horizon d'optimisation de 24 heures en plusieurs sous-horizons.

M me si les erreurs constat es sont relativement faibles, elles peuvent rendre les mod les d'optimisation difficiles   appr hender par les gestionnaires de r seaux si elles ne sont pas mieux prises en compte dans ces mod les. Nous proposons au chapitre suivant de tenir compte des erreurs r siduelles sur la pr vision de la demande.

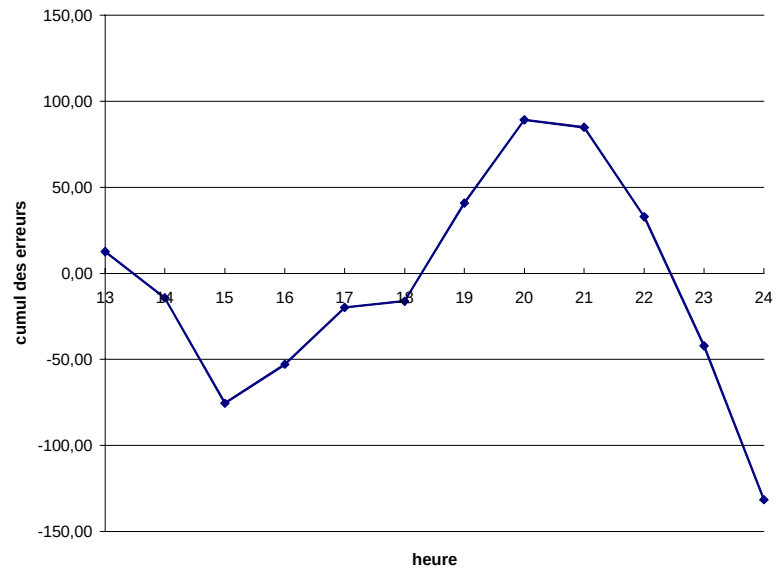


Figure 3.9 : Moyenne des erreurs du modèle à une sortie

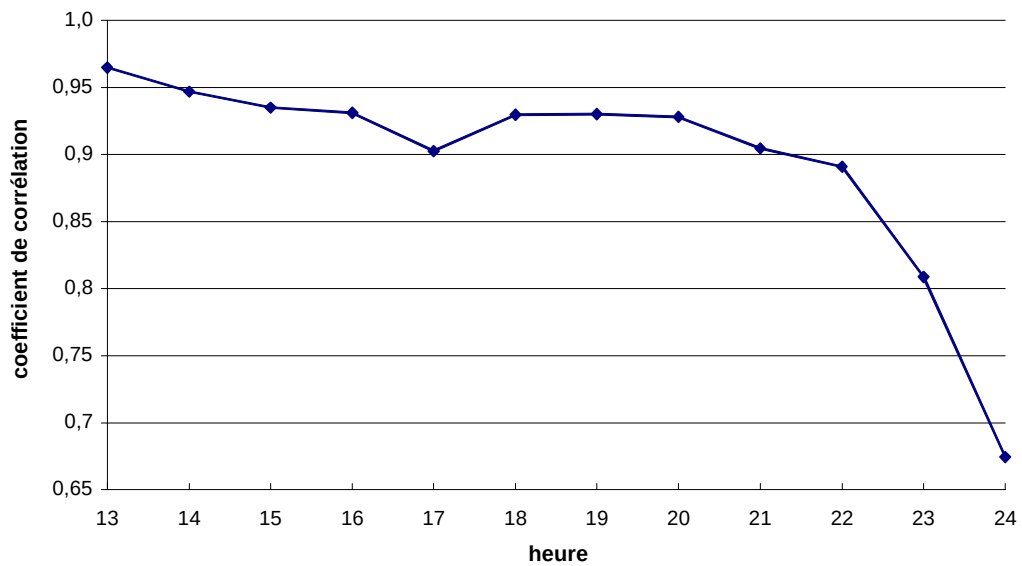


Figure 3.10 : Coefficients de corrélation du modèle à 12 sorties

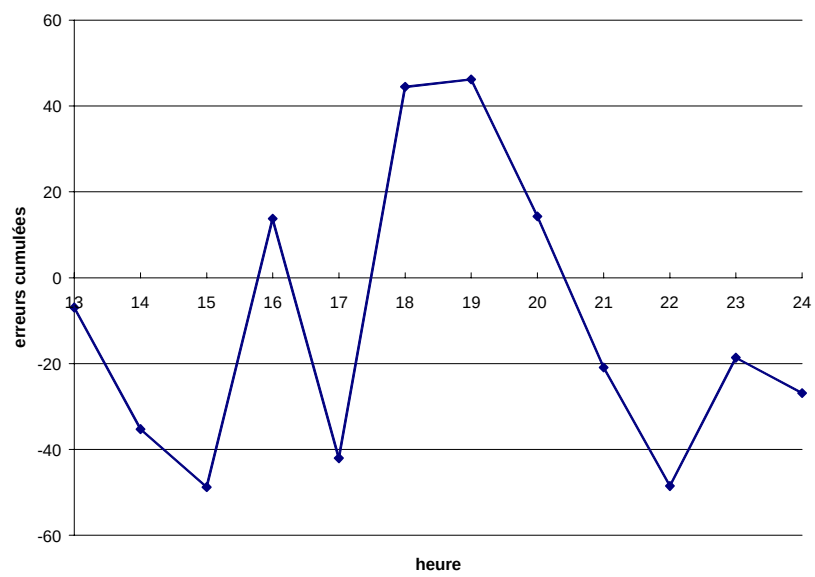


Figure 3.11 : Moyenne des erreurs du mod le   12 sorties

Chapitre 4

Intégration du caractère stochastique de la demande

4.1 Position du problème

Dans un problème de production classique, on cherche à minimiser les coûts en respectant des contraintes de demande et de production.

Si l'on formule ce problème comme un programme linéaire, on pourra l'écrire :

$$\begin{cases} \text{Min}(C, X) \text{ sous} \\ AX = B \end{cases} \quad (4.1)$$

Les éléments de la matrice A et ceux des vecteurs C et B sont supposés connus à l'avance. Dans la réalité, il est souvent difficile d'attribuer à ces paramètres une valeur certaine et il peut être judicieux de les considérer comme des variables aléatoires. Dans les paragraphes qui suivent nous allons examiner les différentes méthodes proposées par Golstein et Youdine dans (Golstein, 1973) afin de résoudre ce type de programme stochastique.

A , B et C ne sont pas toujours considérés aléatoires en même temps. En particulier, si seul le vecteur des coûts est aléatoire et que l'on choisit d'optimiser l'espérance mathématique de la forme linéaire, le programme stochastique se ramène à un programme déterministe où l'on optimise $(\overline{C}, X)$.

Les auteurs appellent programme d'un problème stochastique le type de solution du problème posé.

4.2 Classifications en programmation stochastique

Les auteurs envisagent plusieurs axes pour classer les différents programmes que l'on peut rencontrer.

4.2.1 Classification d'après la notion de programme

4.2.1.1 Formulation serrée

Cette formulation correspond au cas où la violation d'une des contraintes aurait des conséquences très fâcheuses. Le programme est alors constitué par des vecteurs qui satisfont les contraintes dans tous les cas, c'est-à-dire pour toute réalisation q parmi les états de la nature possible Q .

Les contraintes du problème peuvent alors s'écrire :

$$\forall q \in Q, A(q)X \leq B(q), X \geq 0 \quad (4.2)$$

4.2.1.2 Formulation en deux étapes

On commence par fixer le vecteur $X \geq 0$ qui ne vérifie pas nécessairement toutes les contraintes. Dans un deuxième temps, on fixe la réalisation de A et de B . Ceci permet d'introduire ξ (selon la variable $AX - B$) qui corrige la solution adoptée. Le problème s'écrit alors comme la minimisation d'une fonction objectif comportant un terme relatif à la solution adoptée et un terme de pénalité pour non-conformité aux conditions du problème. Si la pénalité devient forte, la formulation tend vers une formulation serrée.

4.2.1.3 Problèmes à contraintes probabilistes

Les m contraintes du problème s'écrivent comme dans l'équation (4.3).

$$\forall i \in \langle 1, m \rangle, P \left(\sum_{j=1}^n a_{ij}x_j \leq b_i \right) \geq p_i \quad (4.3)$$

Dans l'écriture (4.3), les probabilités vérifient naturellement $0 < p_i \leq 1$. Cette écriture signifie que le programme vérifie la $i^{\text{ème}}$ inégalité avec une probabilité P au moins égale à p_i .

4.2.1.4 Programme sous forme de distribution

Il peut être intéressant d'obtenir un vecteur X suivant lui-même une distribution à la manière du vecteur des demandes B . Il faut donc lier X à B . Dans certains problèmes,

il est possible d'établir la relation $X = DB$. Si ce n'est pas le cas, on pose cette relation $X = DB$ et D devient une matrice déterministe inconnue du problème.

4.2.2 Classification d'après le facteur de qualité de la résolution du problème

Les auteurs distinguent 7 critères de qualité que l'on peut retenir pour caractériser le nouvel objectif.

Critère de qualité	Nom du modèle
Espérance mathématique de la forme linéaire (C,X)	E-Modèle
Ecart-type de la forme linéaire (C,X)	V-Modèle
Combinaison linéaire de l'espérance mathématique et l'écart-type de la forme linéaire	EV-Modèle
Probabilité que la forme linéaire dépasse un certain seuil	P-Modèle
Espérance mathématique de la fonction d'utilité de la forme linéaire	U-Modèle
Max-min de la forme linéaire (Max de l'ensemble des X et Min des états de la nature q admissibles)	Mn-Modèle
Max-min de l'espérance mathématique de la forme linéaire	MnE-Modèle

Tableau 4.1 : Classification des programmes en programmation stochastique

4.2.3 Classification suivant l'aspect dynamique de l'incertitude

Les auteurs distinguent les problèmes pour lesquels l'incertitude réside dans les paramètres du problème (la matrice technologique, le coût ou la demande) et ceux pour lesquels l'incertitude vient de fonctions ou de suites aléatoires. Les premiers seront appelés statiques alors que les seconds seront appelés dynamiques.

Dans notre problème nous traitons une incertitude statique, c'est-à-dire portant sur les éléments de A , B ou C de (4.1). L'incertitude dynamique pourrait intervenir si nous voulions utiliser des fonctions de transfert stochastiques comme, par exemple, une distribution sur le transit gravitaire entre deux ouvrages provenant de l'incertitude sur la connaissance de l'état de la conduite. Dans ce dernier cas l'incertitude porterait sur la construction de la matrice A de l'écriture (4.1).

4.3 Détail des méthodes

4.3.1 Formulation serrée

4.3.1.1 Méthode par essai

Dans une première étape, on cherche un vecteur solution, noté $\bar{X}$, du problème déterministe (4.4).

$$\begin{cases} \min (\bar{C}, X) \text{ sous} \\ A^0 X \leq B^0 \\ X \geq 0 \end{cases} \quad (4.4)$$

où $[A^0, B^0]$ est un couple admissible (c'est-à-dire qu'il existe $q > 0$ tel que $A(q) X \leq B(q)$).

Dans un deuxième temps, il faut vérifier que $\bar{X}$ est un "programme permanent" du problème stochastique. Un "programme permanent" est un vecteur qui est solution quel que soit l'état de la nature $q \in Q$ (c'est-à-dire quel que soit le couple admissible $[A, B]$). Ceci peut aussi se traduire par des vecteurs X tels que $P[A(q) X \leq B(q)] = 1$.

Le vecteur $\bar{X}$ est un programme permanent si l'équation (4.5) est vérifiée.

$$\max_{i \in \langle 1; m \rangle} \left[\sum_{j=1}^n a_{ij}^+ \bar{x}_j - b_i^- \right] \leq 0 \quad (4.5)$$

Dans l'équation (4.5) a_{ij}^+ est la borne supérieure des $a_{ij}(q)$ et b_i^- la borne inférieure des $b_i(q)$.

On choisit souvent $A^0 = \bar{A} = EA$ et $B^0 = \bar{B} = EB$

C'est la raison pour laquelle cette méthode s'appelle résolution par rapport aux moyennes.

4.3.1.2 Méthode par intersection des espaces

La méthode précédente ne donne pas forcément de solution car le vecteur solution de la première étape, $\bar{X}$, n'est pas toujours un programme permanent. On propose donc une autre méthode.

Si l'ensemble des couples $[A, B]$ est fini, la résolution du problème stochastique se ramène au problème déterministe (4.6).

$$\begin{cases} \max E(C, X) \text{ sous} \\ \forall q \in \langle 1, k \rangle, A(q) X \leq B(q) \\ X \geq 0 \end{cases} \quad (4.6)$$

Le domaine des admissibles est alors un polyèdre éventuellement non-vidé limité par les k contraintes précédentes.

4.3.2 Problèmes à contraintes probabilistes

4.3.2.1 Méthode 1 : utilisation de la distribution de probabilité

On utilise l'équivalence (4.7).

$$\forall i \in \langle 1; m \rangle, P \left(\sum_{j=1}^n a_{ij} x_j \leq b_i \right) \geq p_i \Leftrightarrow \sum_{j=1}^n a_{ij} x_j \leq \tilde{b}_i \quad (4.7)$$

Dans laquelle $\tilde{b}_i$ est déterminé par $\int_{\tilde{b}_i}^{+\infty} \varphi_i(b_i) db_i = p_i$ et $\varphi_i(b_i)$ est la densité de distribution de la composante b_i

On peut donner une expression de cette densité : $\varphi_i(b_i) = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \varphi(b_1, \dots, b_m) \prod_{j \neq i} db_j$.

Alors le programme stochastique (4.8)

$$\begin{cases} \max E(C, X) \text{ sous} \\ P(AX \leq B) \geq p \end{cases} \quad (4.8)$$

est équivalent au programme déterministe (4.9).

$$\begin{cases} \max (\bar{C}, X) \text{ sous} \\ AX \leq \tilde{B} \end{cases} \quad (4.9)$$

4.3.2.2 Méthode 2 : programme déterministe convexe équivalent

L'idée de cette méthode est de prendre comme inconnue la matrice déterministe D telle que $X = DB$. Le programme stochastique se présente comme (4.10).

$$\begin{cases} \max E(C, X) \text{ sous} \\ P(AX \leq B) \geq p \text{ (les contraintes de positivité sont incluses ici avec } p_i = 1) \\ X = DB \end{cases} \quad (4.10)$$

Ce qui donne pour le facteur de qualité l'expression (4.11).

$$E(C, X) = (\bar{C}, D\bar{B}) = \sum_{i=1}^m \sum_{j=1}^n d_{ij} \bar{b}_i \bar{c}_j \quad (4.11)$$

L'expression (4.11) est entièrement déterministe.

Les contraintes peuvent, elles aussi, prendre des formes déterministes par l'ajout d'inconnues z_i . Elle s'écrivent alors comme détaillé dans (4.12).

$$\begin{cases} (1) a_i D \bar{B} + z_i \leq \bar{b}_i \\ (2) -k_i^2 E [(b_i - \bar{b}_i) - a_i D (B - \bar{B})]^2 + z_i^2 \geq 0 \\ (3) z_i \geq 0 \end{cases} \quad (4.12)$$

La contrainte (2) est obtenue en supposant que le vecteur des contraintes B suit une distribution gaussienne et en utilisant une fonction de Laplace pour la définition de k_i .

La contrainte (1) est linéaire par rapport aux variables d_{ij} et z_i . On peut prouver que les contraintes (2) découpent un espace convexe. Cet espace est concrètement la nappe supérieure, au sens de l'axe des z_i , d'un hyperboloïde à deux nappes. L'intersection des polyèdres définis par (1) et des nappes définies par (2) est un espace convexe. Le problème est donc bien un problème de programmation convexe.

4.3.2.3 Cas des V-Modèle et P-Modèle

On peut également montrer que le traitement des V-Modèle et des P-Modèle aboutit, comme pour les E-Modèle, à un programme déterministe convexe. De la même façon, la fonction objectif est linéaire et les contraintes forment un domaine convexe.

4.3.2.4 Traitement d'un exemple

L'exemple présenté consiste en la production d'un bien sur n périodes. Il s'agit de minimiser l'espérance mathématique de l'ensemble des coûts (production et stockage). Les contraintes sont probabilisées et l'ensemble du programme est présenté en (4.13).

$$\begin{cases} E \left\{ \sum_{j=1}^n (c_j^{(1)} x_j + c_j^{(2)} y_j) \right\} \\ P(x_j + y_j - b_j \geq 0) \geq p_j^{(1)} \text{ (satisfaction de la demande)} \\ P(y_{\min} \leq x_j + y_j - b_j \leq y_{\max}) \geq p_j^{(2)} \text{ (limites de stock)} \\ x_j \geq 0 \\ y_j \geq \begin{cases} x_{j-1} + y_{j-1} - b_{j-1} \\ 0 \end{cases} \text{ (conservation de la masse)} \end{cases} \quad (4.13)$$

Les variables x_j représentent les quantités produites alors que les variables y_j représentent le stock.

Si les volumes produits à une période donnée dépendent de toutes les demandes des périodes précédentes, on peut écrire $x_j = d_{j0} + \sum_{i=1}^{j-1} d_{ij} b_i$. Ceci signifie que la matrice D que l'on introduira sera triangulaire ($d_{ij} = 0$ si $i \geq j$).

4.3.3 Problèmes à deux étapes

On considère un E-Modèle déjà exposé.

$$\begin{cases} \min E(C, X) \text{ sous} \\ AX = B \\ X \geq 0 \end{cases} \quad (4.14)$$

Le principe de cette méthode consiste à choisir $X \geq 0$ solution du programme précédent puis à observer la réalisation de la variable aléatoire B et à procéder à la deuxième étape (4.15).

$$\begin{cases} \min (H, \xi) \text{ sous} \\ D\xi = B - AX \end{cases} \quad (4.15)$$

Le vecteur H et la matrice D sont déterministes et donnés à l'avance. Dans le cas le plus simple, D est la matrice unité et $\xi = \xi^+ - \xi^-$ avec $\begin{cases} \xi^+ = B - AX \text{ et } \xi^- = 0 \text{ si } B \geq AX \\ \xi^+ = 0 \text{ et } \xi^- = AX - B \text{ si } B < AX \end{cases}$

On peut réunir les deux étapes dans un seul programme noté (4.16).

$$\begin{cases} \min_X E \left\{ \min_{\xi} [(C, X) + (H, \xi)] \right\} \\ AX + D\xi = B \\ X \geq 0 \\ \xi \geq 0 \end{cases} \quad (4.16)$$

Cette formulation d'un problème à deux étapes conduit à des problèmes de programmation convexe.

4.3.4 Résolution d'un problème à deux étapes par un programme linéaire déterministe

Dans certains cas, il est possible de se ramener à un programme déterministe linéaire équivalent. Pour présenter cette possibilité, Golstein et Youdine traitent l'exemple d'un réseau de transport.

4.3.4.1 Problème de transport

De façon déterministe le programme s'écrit sous la forme (4.17).

$$\left\{ \begin{array}{l} \text{Min } \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \text{ sous les conditions} \\ (1) \forall j \in \langle 1; n \rangle, \sum_{i=1}^m k_{ij} x_{ij} = b_j \\ (2) \forall i \in \langle 1; m \rangle, \sum_{j=1}^n x_{ij} = a_i \\ (3) \forall i \in \langle 1; m \rangle, \forall j \in \langle 1; n \rangle, x_{ij} \geq 0 \end{array} \right. \quad (4.17)$$

Les contraintes de type (1) traduisent la demande en chacun des B_j alors que les contraintes de type (2) traduisent la capacité de production en chacun des sites A_i .

4.3.4.2 Programme stochastique

Dans l'écriture du programme (4.17) on peut faire intervenir le fait que la demande b_j est aléatoire. Plus précisément, le nombre de ses réalisations est supposé fini. La demande du $j^{\text{ème}}$ produit peut prendre s_j valeurs $b_{j1}, b_{j2}, \dots, b_{js_j}$ avec les probabilités associées $p_{j1}, p_{j2}, \dots, p_{js_j}$.

Les échelons d'accroissement de la demande sont notés de la façon suivante :

$$\forall t \in \langle 2; s_j \rangle, \Delta_{jt} = b_{jt} - b_{j,t-1}, (\Delta_{j1} = b_{j,1})$$

et la probabilité d'une demande supérieure à b_{jt} , $q_{jt} = 1 - \sum_{l=1}^t p_{jl}$. Nous pouvons retrouver la plupart de ces notions sur le graphique de la figure 4.1.

Enfin, on note h_j la pénalité pour l'insatisfaction de la demande par unité du $j^{\text{ème}}$ produit et ξ_j^+ la grandeur de la demande insatisfaite du $j^{\text{ème}}$ produit.

Ainsi, les dépenses totales (frais de transport et insatisfaction de la demande) s'écrivent :

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n h_j \xi_j^+ \quad (4.18)$$

Le problème de distribution stochastique se ramène donc à minimiser l'espérance

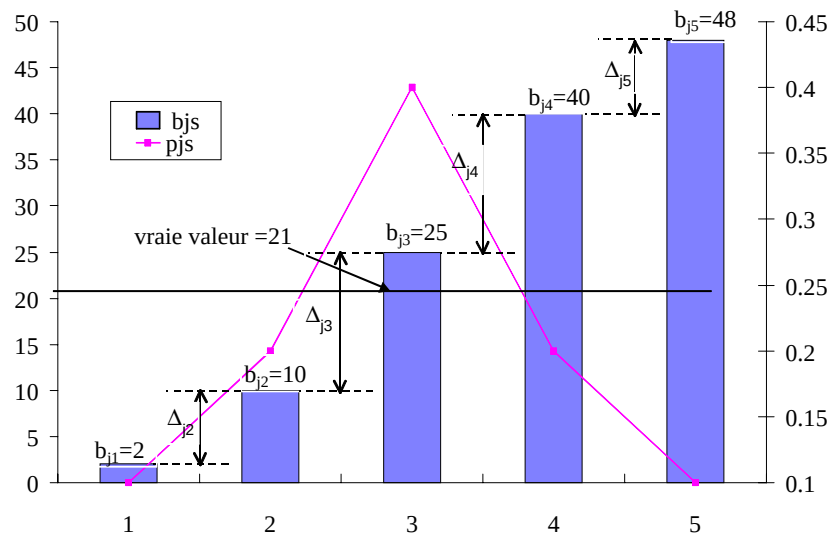


Figure 4.1 : Discretisation de la fonction de demande

mathématique de l'expression (4.18) sous certaines conditions.

$$\left\{ \begin{array}{l} \text{Min } Esp \left(\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n h_j \xi_j^+ \right) \text{ sous} \\ (1) \forall j \in \langle 1; n \rangle, \sum_{i=1}^m k_{ij} x_{ij} + \xi_j^+ - \xi_j^- = b_j \\ (2) \forall i \in \langle 1; m \rangle, \sum_{j=1}^n x_{ij} = a_i \\ (3) \forall i \in \langle 1; m \rangle, \forall j \in \langle 1; n \rangle, x_{ij} \geq 0 \\ (4) \forall j \in \langle 1; n \rangle, \xi_j^+ \geq 0, \xi_j^- \geq 0 \end{array} \right. \quad (4.19)$$

Les nouvelles inconnues sont définies de la façon suivante :

$$\left\{ \begin{array}{l} \xi_j^+ = \begin{cases} b_j - \sum_{i=1}^m k_{ij} x_{ij} & \text{si } b_j \geq \sum_{i=1}^m k_{ij} x_{ij} \\ 0 & \text{sinon} \end{cases} \\ \xi_j^- = \begin{cases} 0 & \text{si } b_j \geq \sum_{i=1}^m k_{ij} x_{ij} \\ \sum_{i=1}^m k_{ij} x_{ij} - b_j & \text{sinon} \end{cases} \end{array} \right. \quad (4.20)$$

4.3.4.3 Programme déterministe équivalent

On peut montrer que le programme stochastique (4.19) est équivalent au programme déterministe (4.21).

$$\left\{ \begin{array}{l} \text{Min } \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} - \sum_{j=1}^n h_j \sum_{s=1}^{s_j} q_{js} \xi_{js} \text{ sous} \\ (1) \forall j \in \langle 1; n \rangle, \sum_{i=1}^m k_{ij} x_{ij} = \sum_{s=1}^{s_j} \xi_{js} \\ (2) \forall i \in \langle 1; m \rangle, \sum_{j=1}^n x_{ij} = a_i \\ (3) \forall i \in \langle 1; m \rangle, \forall j \in \langle 1; n \rangle, x_{ij} \geq 0 \\ (4) \forall j \in \langle 1; n \rangle, \forall s \in \langle 1; s_j \rangle, 0 \leq \xi_{js} \leq \Delta_{js} \end{array} \right. \quad (4.21)$$

Dans le programme (4.21), les ξ_{js} sont des inconnues (tout comme les x_{ij}). Il n'est pas possible d'en donner une expression comme cela était fait pour ξ_j^+ et ξ_j^- .

4.3.4.4 Démonstration de l'équivalence

La démonstration de cette équivalence repose sur les propriétés suivantes des probabilités q_{js} :

- $\forall s < s_j, q_{js} > 0$,
- q_{js} est une fonction décroissante de s .

Ces deux propriétés se traduisent par le fait que si pour la solution du problème (4.21), il existe $r \in \langle 1; s_j \rangle$ tel que $\xi_{jr} \neq 0$ alors $\forall s < r, \xi_{js} = \Delta_{js}$.

On note alors s_j^0 le plus grand indice r tel que $\xi_{jr} \neq 0$, les conditions (1) de (4.19) et de (4.21) s'écrivent alors :

$$\forall j \in \langle 1; n \rangle, \sum_{i=1}^m k_{ij} x_{ij} = \sum_{s=1}^{s_j^0} \xi_{js} \quad (4.22)$$

L'espérance mathématique du coût du problème (4.19) peut s'écrire comme la forme linéaire de (4.23).

$$L = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n h_j \sum_{s=s_j^0}^{s_j} q_{js} \tilde{\xi}_{js} \quad (4.23)$$

Dans l'équation (4.23), on a $\tilde{\xi}_{js_j^0} = \Delta_{js_j^0} - \xi_{js_j^0}$ et $\forall s > s_j^0, \tilde{\xi}_{js} = \Delta_{js}$.

On peut transformer L en retranchant et ajoutant le même terme $\sum_{j=1}^n h_j \sum_{s=1}^{s_j^0} q_{js} \xi_{js}$. L s'écrit alors :

$$L = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n h_j \sum_{s=1}^{s_j} q_{js} \Delta_{js} - \sum_{j=1}^n h_j \sum_{s=1}^{s_j^0} q_{js} \xi_{js} = L_0 + \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} - \sum_{j=1}^n h_j \sum_{s=1}^{s_j^0} q_{js} \xi_{js} \quad (4.2)$$

avec

$$L_0 = \sum_{j=1}^n h_j \sum_{s=1}^{s_j} q_{js} \Delta_{js} \quad (4.25)$$

L_0 peut être interprété comme l'espérance mathématique de la pénalité pour ignorance totale de la demande, lorsque aucune unité de produit n'est livrée aux points de consommation. L_0 ne dépend pas des inconnues et

$$\sum_{j=1}^n h_j \sum_{s=1}^{s_j^0} q_{js} \xi_{js} = \sum_{j=1}^n h_j \sum_{s=1}^{s_j} q_{js} \xi_{js} \text{ car } \forall s > s_j^0, \xi_{js} = 0.$$

Cela signifie que L qui représente l'espérance mathématique du coût du problème déterministe (4.19) ne diffère que par la constante L_0 de la fonction économique du problème déterministe (4.21). On a donc prouvé l'équivalence entre les deux problèmes (4.19) et (4.21).

4.3.4.5 Interprétation de la méthode

Si l'on observe le programme (4.21), on peut tirer quelques conclusions sur la sensibilité aux valeurs des pénalités. En particulier si l'on augmente les pénalités (h_j) pour insatisfaction de la demande, les ξ_{js} vont avoir tendance à croître. Ceci peut alors augmenter à son tour la valeur des variables de décisions (notées ici x_{ij}) par la présence de la contrainte (1).

En résumé, une augmentation de la pénalité h_j à un point de consommation j se traduira par une augmentation de la demande en ce point et donc par une production supplémentaire permettant effectivement de sécuriser le système.

4.4 Utilisation des différentes méthodes et application à un réseau exemple

Nous allons adapter les différentes formulations précédentes à la gestion d'un réseau de distribution d'eau et les appliquer sur le réseau de la figure 4.2. L'application des

méthodes a été mise en oeuvre grâce au solveur Cplex d'Ilog. Nous donnons des temps de calcul indicatifs correspondants aux essais menés sur un Pentium II à 300 Mhz.

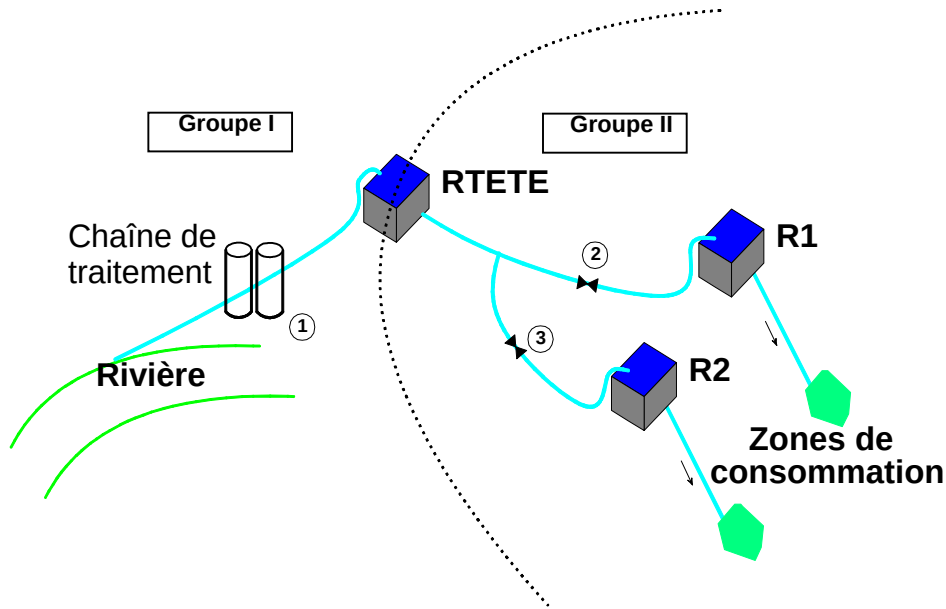


Figure 4.2 : Réseau AEP exemple pour la programmation stochastique

4.4.1 L'aspect stochastique de la demande en eau potable

4.4.1.1 Les modèles de prévision

Les modèles de prévision de la demande que nous avons obtenu peuvent s'écrire $C_r(t) = f_r(\beta(t), \gamma(t)) + \varepsilon_r(t)$. Le terme β regroupe toutes les variables endogènes et le terme γ toutes les variables exogènes utilisées pour la construction du modèle de prévision. Le terme ε_r représente l'erreur de prévision. C'est pour tenir compte de ce terme que nous voulons introduire la programmation stochastique. La distribution des termes ε_r est aléatoire de moyenne nulle et d'écart-type donné $\sigma(\varepsilon_r)$.

On peut, bien entendu, regrouper les deux termes et présenter alors la demande $C_r(t)$ comme ayant une distribution gaussienne de moyenne $\mu(C_r(t))$ et d'écart-type $\sigma(\varepsilon_r)$.

4.4.1.2 Distribution de la demande

Pour chaque variable aléatoire $C_r(t)$, nous noterons $\mu(C_r(t))$ la moyenne et $\sigma(C_r(t)) = \sigma(\varepsilon_r)$ l'écart-type. Alors, les sommes partielles $C_{r\theta} = \left(\sum_{t=1}^{\theta} C_r(t) \right)$ suivent également une

distribution gaussienne de moyenne $\mu(C_{r\theta}) = \left(\sum_{t=1}^{\theta} \mu(C_r(t)) \right)$ et d'écart-type $\sigma(C_{r\theta}) = \left(\sqrt{\sum_{t=1}^{\theta} \sigma^2(C_r(t))} \right)$. Dans toutes les méthodes que nous proposons ce sont bien ces sommes partielles qui constitueront nos données aléatoires.

Dans certaines méthodes (formulation serrée et formulation en deux étapes), nous aurons besoin de discrétiser la distribution de la demande. Dans ce cas, nous utiliserons les propriétés d'une distribution gaussienne rappelées figure 4.3.

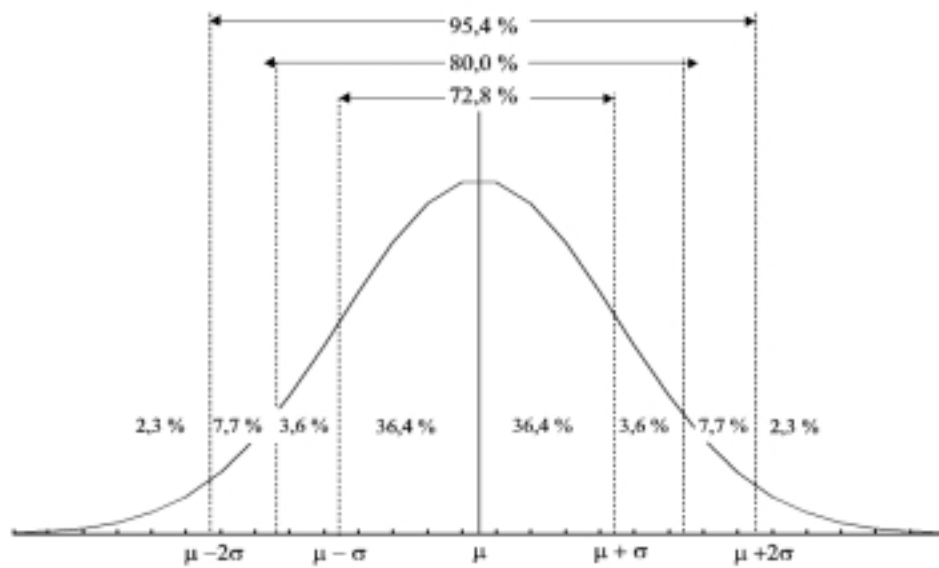


Figure 4.3 : Propriété d'une distribution gaussienne

4.4.1.3 Le réseau exemple

Le réseau présenté figure 4.2 comporte trois organes de commande qui ont été séparés en deux groupes indépendants grâce à la séparation hydraulique réalisée par RTETE. Deux réservoirs du réseau alimentent une zone de consommation. A chacune de ces zones, nous associons la valeur de la consommation à chaque pas de temps ainsi que la valeur estimée de l'écart-type de cette consommation.

4.4.2 Formulation serrée

On peut considérer qu'il est impératif de ne jamais avoir de réservoirs vides ou qui débordent. C'est pourquoi, la formulation serrée d'un problème stochastique répond bien

à notre problème.

Nous noterons :

- $s_{r\theta}$, le nombre de pas de discrétisation pour chaque réservoir et à chaque pas de temps,
- $\forall s \in \langle 1; s_{r\theta} \rangle$, $C_{r\theta s}$, la valeur discrète de la demande aléatoire cumulée $C_{r\theta}$.

Si nous utilisons comme discrétisation les bornes représentées sur le graphique de la figure 4.3, nous aurions : $s_{r\theta} = 5$ et, en particulier,

$C_{r\theta 1} = \mu(C_{r\theta}) - 2\sigma(C_{r\theta})$ et $C_{r\theta 5} = \mu(C_{r\theta}) + 2\sigma(C_{r\theta})$. Pour nous approcher davantage d'une probabilité de 100 %, nous choisissons de rajouter une valeur et de prendre $s_{r\theta} = 6$ avec $C_{r\theta 6} = \mu(C_{r\theta}) + 4\sigma(C_{r\theta})$. Plus précisément, ce choix nous placera à une probabilité de 99,997 %.

4.4.2.1 Ecriture du programme

$$\left\{ \begin{array}{l} z_{\min} = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) \\ (1) \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, \forall s \in \langle 1; s_{r\theta} \rangle, \\ V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} \leq C_{r\theta s} \\ (2) \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, \forall s \in \langle 1; s_{r\theta} \rangle, \\ V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\min} \geq C_{r\theta s} \\ (3) \forall r \in \langle 1; R \rangle, \forall s \in \langle 1; s_{rT} \rangle, V_r(0) + \sum_{t=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} = C_{r24s} \\ (4) \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \\ (5) \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \forall i \in \langle 1; I_g \rangle, \alpha_i^g(\theta) \geq 0 \end{array} \right. \quad (4.26)$$

4.4.2.2 Contraintes incompatibles

La méthode de formulation serrée peut aboutir à un problème infaisable. Le domaine des admissibles constitué par les différentes contraintes peut, en effet, être vide.

Dans notre cas, les contraintes de type (3) du programme (4.26) sont incompatibles entre elles. Ceci est dû à la présence d'égalités. Nous pouvons alors remplacer cette contrainte (3) par l'expression (4.27).

$$\forall r \in \langle 1; R \rangle, \forall s \in \langle 1, s_{rT} \rangle, V_r(0) + \sum_{t=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^T C_{rts} \geq V_{r\max} \quad (4.27)$$

Cette façon d'écrire la contrainte reste cohérente avec une formulation serrée puisque même dans le cas le plus défavorable, la contrainte (4.27) imposera un réservoir plein en fin d'horizon. Plus précisément, nous aurons des contraintes qui imposeront un débordement des réservoirs. Dans la pratique, nous ne transmettrons pas toutes les commandes mais le calcul ainsi effectué nous donnera un majorant de la solution recherchée en respectant toutes les contraintes.

Toutefois, une autre incompatibilité peut alors exister. Il s'agit de celle entre les contraintes (1) de (4.26) et les contraintes (4.27). Les contraintes (1) de (4.26) imposent, en effet, que le volume contenu dans les réservoirs reste inférieur au volume maximal au cours des $(T - 1)$ premiers pas de temps alors que les contraintes (4.27) que nous venons d'écrire impose que ce volume soit supérieur ou égal au volume maximal au dernier pas de temps. En fonction des capacités d'entrée dans le réservoir et des consommations au cours du dernier pas de temps, le problème pourra ne pas avoir de solution. Dans ce cas nous choisirons de relâcher les contraintes de type (1) de (4.26) sur l'avant dernier pas de temps $(T - 1)$. Le cas échéant, nous pourrions aussi relâcher les contraintes de type (1) sur plusieurs pas de temps à la fin de l'horizon.

4.4.2.3 Application

Sur notre exemple nous rencontrons effectivement des problèmes d'incompatibilité sur les contraintes (1). Par essais successifs, nous obtenons une solution en réduisant le domaine d'application de la contrainte (1) à l'intervalle $\theta \in \langle 1; T - 3 \rangle$.

Le temps de calcul est alors de 0,52 sec CPU. La valeur de la fonction objectif servira de majorant à l'ensemble de notre problème. Nous noterons donc 100 cette valeur.

4.4.3 Formulation de contraintes probabilistes

On peut sans doute accepter de relâcher un peu le système exposé au paragraphe précédent. Au lieu de considérer qu'aucune contrainte ne peut être violée, il peut être plus judicieux de probabiliser les différentes contraintes.

4.4.3.1 Présentation probabilisée

$$\left\{ \begin{array}{l}
z_{\min} = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) \\
(1) \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, \\
P \left(V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} \leq C_{r\theta} \right) \geq p_{r\theta}^1 \\
(2) \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, \\
P \left(V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\min} \geq C_{r\theta} \right) \geq p_{r\theta}^2 \\
(3) \forall r \in \langle 1; R \rangle, P \left(V_r(0) + \sum_{t=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} = C_{rT} \right) \geq p_r^3 \\
(4) \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \\
(5) \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \forall i \in \langle 1; I_g \rangle, \alpha_i^g(\theta) \geq 0
\end{array} \right. \quad (4.28)$$

4.4.3.2 Transformation en programme déterministe

La contrainte (1) du programme (4.28) est bien dans le cadre de l'équivalence que nous voulons proposer puisque l'on peut écrire la relation .

$$\begin{aligned}
& \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, P \left(V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} \leq C_{r\theta} \right) \geq p_{r\theta}^1 \\
& \Leftrightarrow V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} \leq \overline{C_{r\theta}}
\end{aligned} \quad (4.29)$$

Dans cette équation $\overline{C_{r\theta}}$ défini par l'équation (4.30).

$$\int_{\overline{C_{r\theta}}}^{+\infty} \varphi_{r\theta}(C_{r\theta}) dC_{r\theta} = p_{r\theta}^1 \quad (4.30)$$

Dans le cadre d'une distribution gaussienne la densité de distribution est bien connue et il est aisé d'inverser l'équation (4.30).

Par exemple, si $p_{r\theta}^1 = 0,9$, $\overline{C_{r\theta}} = \mu(C_{r\theta}) - 1,41 \times \sigma(C_{r\theta})$.

Dans (4.28), la contrainte (3) peut se présenter comme la contrainte (2) car on peut écrire :

$$(3) \forall r \in \langle 1; R \rangle, P \left(V_r(0) + \sum_{t=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} \geq C_{rT} \right) \geq p_r^3 \quad (4.31)$$

La transformation en contraintes déterministes se fait de façon symétrique par rapport à la transformation de la contrainte (1). Voyons en 4.32 l'équivalence pour la

contrainte (2).

$$\begin{aligned} & \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, P \left(\sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r \min} + V_r(0) \geq C_{r\theta} \right) \geq p_{r\theta}^2 \\ & \Leftrightarrow \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r \min} + V_r(0) \geq \overline{\overline{C_{r\theta}}} \end{aligned} \quad (4.32)$$

avec $\overline{\overline{C_{r\theta}}}$ défini par $\int_{-\infty}^{\overline{\overline{C_{r\theta}}}} \varphi_{r\theta}(C_{r\theta}) d(C_{r\theta}) = p_{r\theta}^2$. Comme précédemment, nous pouvons prendre comme exemple $p_{r\theta}^2 = 0,9$. On a alors : $\overline{\overline{C_{r\theta}}} = \mu(C_{r\theta}) + 1,41 \times \sigma(C_{r\theta})$.

La différence de signe devant le coefficient 1,41 dans les expressions de $\overline{C_{r\theta}}$ et de $\overline{\overline{C_{r\theta}}}$ s'explique par le sens contraire des inégalités à l'intérieur de la fonction de probabilité P des contraintes (1) et (2) de (4.28).

4.4.3.3 Application

Pour s'assurer de la cohérence de nos calculs, nous pouvons construire un modèle pour lequel toutes les probabilités valent 99,997 %. Il nous faut là aussi relâcher les contraintes de stock maximal dans les derniers pas de temps de l'horizon d'optimisation. Ceci donne le même résultat que la formulation serrée.

Une fois cette vérification effectuée, nous modifions les valeurs des différentes probabilités. Nous présentons le résultat des différentes simulations dans le tableau 4.2.

	$p_{r\theta 1} = p_{r\theta 2} = 99,997\%$	$p_{r\theta 1} = p_{r\theta 2} = 90\%$
	C=100	C=100
$p_{r3} = 99,997\%$	Contrainte (1) utilisable de 1 à 20 h	Contrainte (1) utilisable de 1 à 22 h
	C=95,05	C=95,05
$p_{r3} = 90\%$	Contrainte (1) utilisable de 1 à 22 h	Contrainte (1) utilisable de 1 à 22 h

Tableau 4.2 : Résultats de l'optimisation probabilisée

Nous pouvons en conclure que le choix de la probabilité de remplissage des réservoirs a une influence forte sur le coût de la solution. Le choix des probabilités liées aux niveaux minimum et maximum dans les réservoirs ne modifie pas les coûts dans notre exemple mais peut éventuellement permettre de rencontrer moins de problèmes d'incompatibilité.

Les temps de calcul de ces différentes simulations sont très proches les uns des autres et avoisinent les 0,5 sec.

4.4.4 Formulation en problèmes à deux étapes

4.4.4.1 Ecriture du programme stochastique

Il s'agit, dans un premier temps, de modifier le programme linéaire (4.33), dans lequel nous plaçons les demandes en second membres, afin d'en proposer une formulation proche de (4.19).

$$\left\{ \begin{array}{l} z_{\min} = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) \\ (1) \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} \leq \sum_{t=1}^{\theta} C_r(t) \\ (2) \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, V_{r\min} - V_r(0) - \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) \leq - \sum_{t=1}^{\theta} C_r(t) \\ (3) \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} = \sum_{t=1}^T C_r(t) \\ (4) \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \\ (5) \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \forall i \in \langle 1; I_g \rangle, \alpha_i^g(\theta) \geq 0 \end{array} \right. \quad (4.33)$$

Deux aspects du programme (4.33) diffèrent de ceux de (4.19) :

- les contraintes (1) et (2) de (4.33) dans lesquelles les fonctions de demande, $C_r(t)$, interviennent sont des inégalités. Ces contraintes ne traduisent plus une demande qu'il faut satisfaire mais l'existence de niveau maxi et mini dans chacun des réservoirs.
- les fonctions de demande apparaissent sous la forme de sommes partielles du type

$$\forall \theta \in \langle 1; T \rangle, C_{r\theta} = \sum_{t=1}^{\theta} C_r(t)$$

Nous pouvons rapidement régler la deuxième de ces deux différences. En effet, la méthode pourra s'appliquer en faisant des sommes de variables d'écart sans que cela ne modifie le principe.

La première différence est plus délicate à transformer. D'un point de vue pratique, notre problème ne doit plus se voir comme un risque d'avoir une demande insatisfaite ou une production superflue. Il s'agit plutôt de rencontrer le risque d'un débordement d'un réservoir ou, cas plus grave, d'avoir un réservoir vide.

Néanmoins, c'est bien la demande qui est aléatoire. C'est pourquoi, nous définirons, comme dans le modèle de Golstein et Youdine, $\forall r \in \langle 1; 3 \rangle, \forall \theta \in \langle 1; 24 \rangle, \xi_{r\theta}^+$ et $\xi_{r\theta}^-$ les écarts à la demande moyenne et $h_{r\theta}$ la pénalité pour écart positif (c'est-à-dire pour une demande supérieure à ce qui était prévu). Cette pénalité traduit indirectement le risque

de voir le niveau d'un réservoir descendre en dessous de son niveau mini. Ce risque n'existera réellement que lorsque le niveau au début du pas de temps θ est effectivement proche du niveau minimal.

Le programme stochastique pourra alors s'écrire comme (4.34).

$$\left\{ \begin{array}{l} z_{\min} = Esp \left(\sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) + \sum_{r=1}^R \sum_{\theta=1}^T h_{r\theta} \xi_{r\theta}^+ \right) \\ (1) \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, \\ V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} - \sum_{t=1}^{\theta} \xi_{rt}^+ + \sum_{t=1}^{\theta} \xi_{rt}^- \leq \sum_{t=1}^{\theta} C_r(t) \\ (2) \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, \\ V_{r\min} - V_r(0) - \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) + \sum_{t=1}^{\theta} \xi_{rt}^+ - \sum_{t=1}^{\theta} \xi_{rt}^- \leq - \sum_{t=1}^{\theta} C_r(t) \\ (3) \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} - \sum_{t=1}^T \xi_{rt}^+ + \sum_{t=1}^T \xi_{rt}^- = \sum_{t=1}^T C_r(t) \\ (4) \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \\ (5) \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \forall i \in \langle 1; I_g \rangle, \alpha_i^g(\theta) \geq 0 \\ (6) \forall \theta \in \langle 1; T \rangle, \forall r \in \langle 1; R \rangle, \xi_{r\theta}^+ \geq 0, \xi_{r\theta}^- \geq 0 \end{array} \right. \quad (4.34)$$

4.4.4.2 Programme déterministe équivalent

Nous considérerons par la suite que le nombre de réalisations de chaque $C_r(\theta)$ est fini. $C_r(\theta)$ pourra prendre $s_{r\theta}$ valeurs notées $C_{r\theta s}$ avec $s = 1, 2, \dots, s_{r\theta}$ avec les probabilités associées $p_{r\theta s}$.

En poursuivant les analogies avec le passage de (4.19) à (4.21), nous noterons $q_{r\theta m}$ la probabilité qu'une valeur des demandes soit supérieure à $C_{r\theta m}$ ($q_{r\theta m} = 1 - \sum_{l=1}^m p_{r\theta l}$). De plus, nous notons $\forall m \in \langle 2; s_{r\theta} \rangle, \Delta_{r\theta m} = C_{r\theta m} - C_{r\theta(m-1)}$ ($\Delta_{r\theta,1} = C_{r\theta,1}$) les échelons d'accroissement de $C_{r\theta m}$.

En utilisant les propriétés d'une distribution gaussienne, nous pourrions attribuer les valeurs discrètes et les probabilités à la variable $C_r(\theta)$. Ces valeurs figurent dans le tableau 4.3.

valeur de $C_r(\theta)$	Probabilité de consommation supérieure à $C_r(\theta)$
$C_{r\theta 1} = \bar{C}_{r\theta} - 2\sigma(C_{r\theta})$	$q_{r\theta 1} = 0.9772$
$C_{r\theta 2} = \bar{C}_{r\theta} - \sigma(C_{r\theta})$	$q_{r\theta 2} = 0.8413$
$C_{r\theta 3} = \bar{C}_{r\theta}$	$q_{r\theta 3} = 0.5000$
$C_{r\theta 4} = \bar{C}_{r\theta} + \sigma(C_{r\theta})$	$q_{r\theta 4} = 0.1587$
$C_{r\theta 5} = \bar{C}_{r\theta} + 2\sigma(C_{r\theta})$	$q_{r\theta 5} = 0.0228$

Tableau 4.3 : Valeurs de la demande discrète

Le programme déterministe équivalent au programme (4.34) s'écrit alors (4.35).

$$\left\{ \begin{array}{l}
 z_{\min} = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) - \sum_{r=1}^R \sum_{\theta=1}^T h_{r\theta} \sum_{s=1}^{s_{r\theta}} q_{r\theta s} \xi_{r\theta s} \\
 (1) \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, \\
 V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} \leq \sum_{t=1}^{\theta} \sum_{s=1}^{s_{rt}} \xi_{rts} \\
 (2) \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, \\
 V_{r\min} - V_r(0) - \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) \leq - \sum_{t=1}^{\theta} \sum_{s=1}^{s_{rt}} \xi_{trs} \\
 (3) \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} = \sum_{t=1}^T \sum_{s=1}^{s_{rt}} \xi_{trs} \\
 (4) \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \\
 (5) \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \forall i \in \langle 1; I_g \rangle, \alpha_i^g(\theta) \geq 0 \\
 (6) \forall \theta \in \langle 1; T \rangle, \forall r \in \langle 1; R \rangle, \forall s \in \langle 1; s_{r\theta} \rangle, 0 \leq \xi_{r\theta s} \leq \Delta_{r\theta s}
 \end{array} \right. \quad (4.35)$$

L'équivalence entre les programmes (4.34) et (4.35) se démontre comme celle entre (4.19) et (4.21) présentée au paragraphe 4.3.4.4. La présence d'inégalités ne change pas le raisonnement puisque les deux contraintes (1) de (4.34) et (4.35) peuvent toutes deux s'écrire :

$$\forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - V_{r\max} \leq \sum_{t=1}^{\theta} \sum_{s=1}^{s_{rt}^0} \xi_{rts} \quad (4.36)$$

avec s_{rt}^0 l'indice tel que $\forall s_{rt} > s_{rt}^0, \xi_{rts} = 0$. Le même raisonnement peut bien-sûr être mené pour les contraintes de type (2).

4.4.4.3 Application

Le temps de calcul est très court, moins de 1 sec CPU. Le plus intéressant à détailler dans les calculs réels que nous avons mis en oeuvre est la sensibilité aux valeurs de

pénalité retenues. Nous avons déjà écrit au paragraphe 4.3.4.5 ce qui pouvait être attendu d'une telle méthode.

Nous pouvons d'abord examiner les consommations utilisées par le modèle lorsque nous appliquons une pénalité variable sur 24 heures. Nous avons fait cette simulation sur R2 en prenant des pénalités variant de $0.05 F/m^3$ à $10 F/m^3$ au cours de la journée. Nous obtenons alors les consommations représentées figure 4.4.

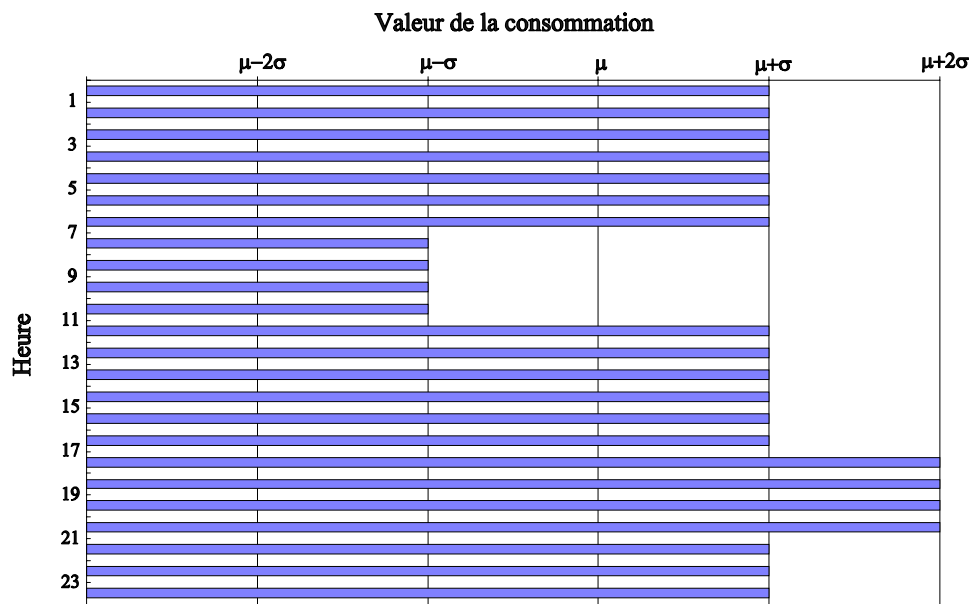


Figure 4.4 : Exemple de consommations dans une formulation en deux étapes

Par ailleurs, nous avons fait trois simulations pour lesquelles

$\forall \theta \in \langle 1; 24 \rangle, h_{1,\theta} = 5 F/m^3$. Pour chacune de ces trois simulations nous avons :

- $\forall \theta \in \langle 1; 24 \rangle, h_{2,\theta} = 0.05 F/m^3$
- $\forall \theta \in \langle 1; 24 \rangle, h_{2,\theta} = 0.1 F/m^3$
- $\forall \theta \in \langle 1; 24 \rangle, h_{2,\theta} = 50 F/m^3$

Nous pouvons tout d'abord comparer le coût des différentes simulations ainsi que la consommation totale qui a été affectée au réservoir R2 sur le graphique 4.4.

Pénalités (F/m3)	0.05	0.10	50.00	déterministe
Conso R2 (m3/j)	384.00	428.00	511.11	428.00
Coût	99.2	103.6	112.0	100

Tableau 4.4 : Influence du paramètre pénalité sur le coût et la consommation

Comme cela était prévu, plus la pénalité est importante plus la consommation journalière affectée au réservoir est forte. Ces résultats sont à comparer avec les 428 m^3 affectés au réservoir R2 lorsque nous simulons la situation déterministe. Pour bien visualiser cet écart par rapport à la situation déterministe, voyons ce que serait le niveau dans R2 au cours de l'optimisation si la consommation réellement constatée correspondait finalement à la moyenne de la distribution utilisée.

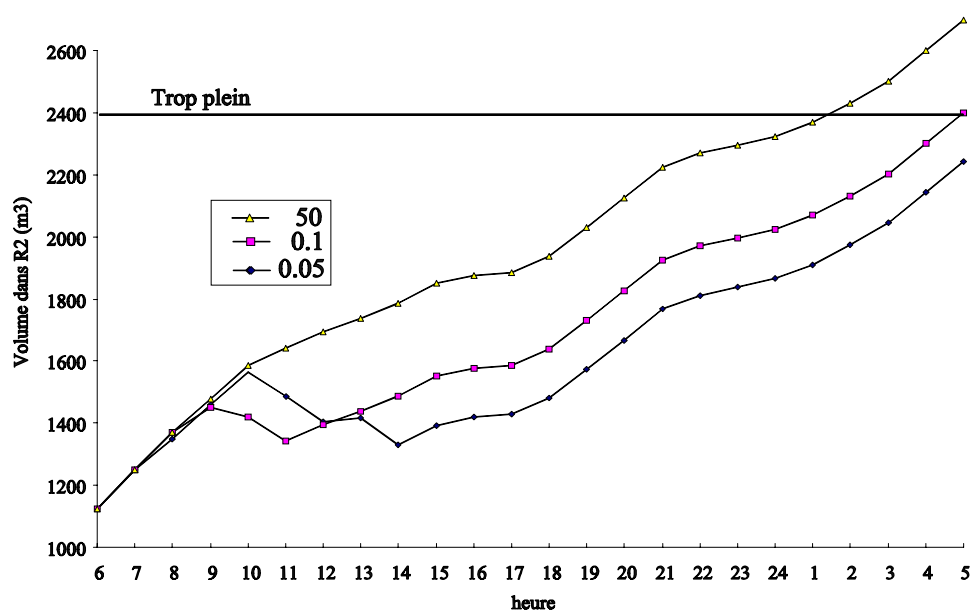


Figure 4.5 : Trajectoires obtenues en faisant varier le risque

On voit sur le graphique de la figure 4.5 que les pénalités fortes entraînent une sécurité plus importante mais augmentent également le risque d'un débordement du réservoir en fin d'horizon si la consommation réellement mesurée est égale à la moyenne de la distribution.

4.5 Conclusion

L'objectif d'intégration des erreurs de prévision de la demande dans les modèles d'optimisation est atteint par plusieurs méthodes dont la philosophie est différente. L'idée naturelle de départ consistant à vouloir transformer la distribution de demande pour aboutir à une distribution des commandes à passer sur le réseau n'est pas compatible avec la complexité de notre problème. Par contre, si nous acceptons de discrétiser la distribution de la demande voire de ne tenir compte que des extrêmes de cette distribu-

tion, nous pouvons écrire un modèle d'optimisation plus complet pour lequel la sécurité d'approvisionnement des abonnés devient un paramètre choisi par le gestionnaire du réseau.

Chapitre 5

Prise en compte de contraintes de fonctionnement

5.1 Position du problème

Les contraintes de fonctionnement que nous voulons intégrer ici sont ce que l'on peut également appeler des contraintes de continuité. Elles consistent à imposer sur un ouvrage le principe suivant : "toute période de marche de l'ouvrage doit avoir une durée minimale". Cette contrainte apparaît notamment sur des usines de traitement d'eau potable où un fonctionnement par à coups peut entraîner des problèmes de qualité de l'eau traitée.

La difficulté du traitement de ce type de contraintes vient du fait qu'elle impose une contradiction au principe d'optimalité de Bellman. Celui-ci s'énonce de la façon suivante "dans une trajectoire optimale, toute sous-trajectoire est également optimale". Ceci signifie que, même en programmation dynamique, ce type de contraintes ne peut être traité directement par une méthode de programmation mathématique.

Dans certains cas pratiques, il ne s'agira plus de respecter une contrainte aussi stricte que celle énoncée plus haut mais plutôt de chercher à minimiser le nombre de démarrage de pompes ou d'ouverture de vannes afin de minimiser l'usure de ces organes.

5.2 Utilisation des graphes

La programmation linéaire par les graphes va nous permettre d'illustrer les modifications apportées par le fait d'imposer une contrainte de continuité sur un ouvrage.

5.2.1 Représentation de la contrainte de continuité

Les arcs sur lesquels la contrainte de continuité va pouvoir être représentée sont ceux du type "arcs de transit" et de la famille "rivière-bâche". Leur capacité est donc égale à la capacité horaire de l'usine de traitement d'eau potable (voir figure 2.3) et leur coût vaudra le coût de production horaire de cette usine (incluant les coûts électriques et les coûts de traitement). Pour que la contrainte de continuité soit respectée, il faudra que les arcs soient saturés par périodes continues. Pour fixer les idées, nous prenons comme description de la contrainte de continuité de l'usine de traitement d'eau potable que celle-ci doit fonctionner par plages d'une durée minimale de 4 heures entrecoupées de période d'arrêt d'au moins 1 heure. Si l'on regarde uniquement les arcs représentant le fonctionnement de l'usine, on obtient des solutions du type de celle représentée figure 5.1.

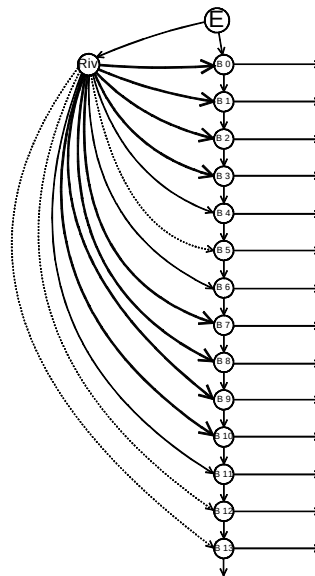


Figure 5.1 : Graphe d'une solution respectant la contrainte de continuité

Sur les figures 5.1 les arcs représentant le fonctionnement de l'usine peuvent être représentés de trois façons différentes :

- en trait gras pour signifier que l'arc est saturé,
- en trait pointillé pour signifier que le flot est nul sur l'arc,
- en trait fin et plein pour les cas où le flot est strictement inférieur à la capacité et non nul.

La solution représentée figure 5.1 n'est qu'un exemple d'une solution respectant la contrainte de continuité de l'usine. Il peut y avoir de 1 à 5 plages de fonctionnement

sur l'horizon d'optimisation et ces plages peuvent, suivant leur nombre, durer de 4 à 24 heures. Pour un dénombrement précis du nombre de scénarios respectant les contraintes de continuité évoquées précédemment, on pourra se reporter à (Guhl, 1995).

5.2.2 Méthodes de résolution

Même si l'algorithme de résolution du flot maximal à coût minimal est très rapide, il est apparu que la méthode consistant à examiner de façon systématique tous les scénarios possibles et à retenir celui de coût minimal donnait des temps de calcul incompatible avec notre volonté d'utiliser les méthodes mises en place pour la gestion en temps réel des réseaux d'eau potable. Pour un réseau comportant une dizaine de réservoirs et autant d'organes de commande et sur lequel nous imposons une seule contrainte de continuité, nous pouvons atteindre des temps de calcul de 4 heures.

Nous nous sommes alors orientés vers deux types de méthodes différentes pour trouver un algorithme exhibant une solution respectant la contrainte de continuité tout en gardant des temps de calcul compatibles avec la gestion en temps réel des ouvrages :

1. La modification artificielle des coûts sur les arcs représentant l'usine. Ce type de méthode a l'avantage de donner des temps de calcul très faibles puisqu'elle ne donne lieu qu'à un calcul de flot maximal à coût minimal.
2. L'examen non exhaustif des scénarios de fonctionnement de l'usine respectant les contraintes de continuité. Il faut pour cela trouver une règle de limitation du nombre de scénarios à calculer.

La contrainte de continuité étant à respecter de façon stricte, nous avons opté pour le deuxième type de méthodes, le premier ne permettant pas de garantir ce respect. Il a alors fallu trouver un critère de limitation du nombre de scénarios à étudier. Nous avons choisi une méthode en deux temps. Un premier calcul de flot maximal à coût minimal effectué à partir du graphe initial de la figure 2.3 nous permet de trouver le temps de fonctionnement de l'usine total sur l'horizon d'optimisation sans respect de la contrainte de continuité. Nous recherchons alors, dans un deuxième temps, la solution optimale respectant la contrainte de continuité parmi les scénarios dont la durée totale du temps de fonctionnement de l'usine sur l'horizon d'optimisation est proche de la valeur initiale.

5.2.3 Résultats

Les résultats fournis ici résultent de l'application des modèles présentés au réseau du Syndicat Intercommunal des Eaux de la Région de Sarralbe. Nous allons comparer trois

calculs :

1. un calcul sans prise en compte de la contrainte de continuité que nous appellerons CLASS,
2. un calcul examinant tous les scénarios possibles, appelé INTEG,
3. un calcul où l'on examine que les scénarios dont le temps de fonctionnement de l'usine sur l'horizon est proche de celui obtenu pour le programme CLASS, appelé TDF.

Voyons dans le tableau 5.1 les résultats de notre comparaison.

Méthode employée	Temps de calcul (s)	Coût total de la solution (valeur arbitraire)
CLASS	2	100
INTEG	17460	100
TDF	59	104,3

Tableau 5.1: Comparaison des trois méthodes de résolution de la contrainte de continuité par les graphes

En plus des temps de calcul qui sont conformes à ce que nous avons déjà indiqué, il est intéressant de noter les différences de coûts qui apparaissent dans le tableau 5.1. Nous pouvons noter, d'une part, que le fait d'imposer la contrainte de continuité n'augmente pas, sur notre exemple, le coût total de la trajectoire obtenue. Nous pouvons remarquer, d'autre part, que le fait d'avoir retenu une méthode simplifiée peut entraîner un surcoût (de 4,3 % ici).

Voyons maintenant la traduction de la contrainte de continuité sur le fonctionnement de l'usine au cours de l'horizon d'optimisation. Pour cela nous pouvons comparer les représentations des figures 5.2 et 5.3.

Si l'on compare les graphiques des figures 5.2 et 5.3 on peut constater que le fonctionnement de l'usine durant les heures de nuit est le même. Ces heures étant moins coûteuses (du fait des tarifs électriques heures creuses), l'optimisation conduit dans les deux cas à un fonctionnement continu de l'usine au cours de la période 22 heures-6 heures. La différence a lieu sur le fonctionnement au cours des heures pleines de tarifs électriques. Au cours de l'optimisation ne tenant pas compte des contraintes de continuité, le fonctionnement de l'usine se fait par petites périodes au cours de cette plage horaire. Lorsque la contrainte de continuité est imposée, l'ensemble de ces périodes est regroupé sur une seule plage répondant aux contraintes de continuité. C'est cet aspect regroupement qui explique le fait que les deux trajectoires aient le même coût. Ceci n'est

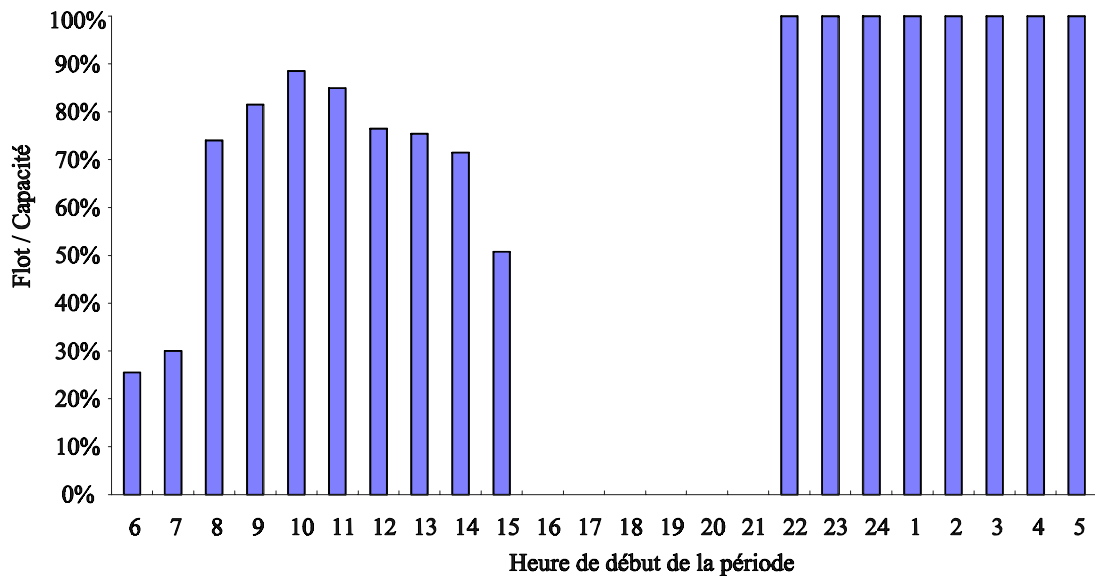


Figure 5.2 : Fonctionnement de l'usine sans contraintes de continuité

pas à généraliser. Il existe des situations où le fait d'imposer une contrainte de continuité augmentera le coût de la solution soit en augmentant le temps de fonctionnement global de l'usine sur l'horizon d'optimisation soit en déplaçant des périodes de fonctionnement vers des heures de coût électrique plus cher.

5.3 Utilisation de la programmation linéaire en nombres entiers

5.3.1 Modèle sans respect de la contrainte de continuité

Le modèle d'optimisation en programmation linéaire que nous utilisons a été présentée à l'équation (2.9).

5.3.2 Continuité de fonctionnement

5.3.2.1 Méthode d'implication

Nous aurons besoin de faire intervenir des implications dans notre programme linéaire. Pour cela, voyons une méthode utilisant des variables binaires. Soit l'implication suivante : $f(x) > 0 \Rightarrow g(x) \geq 0$

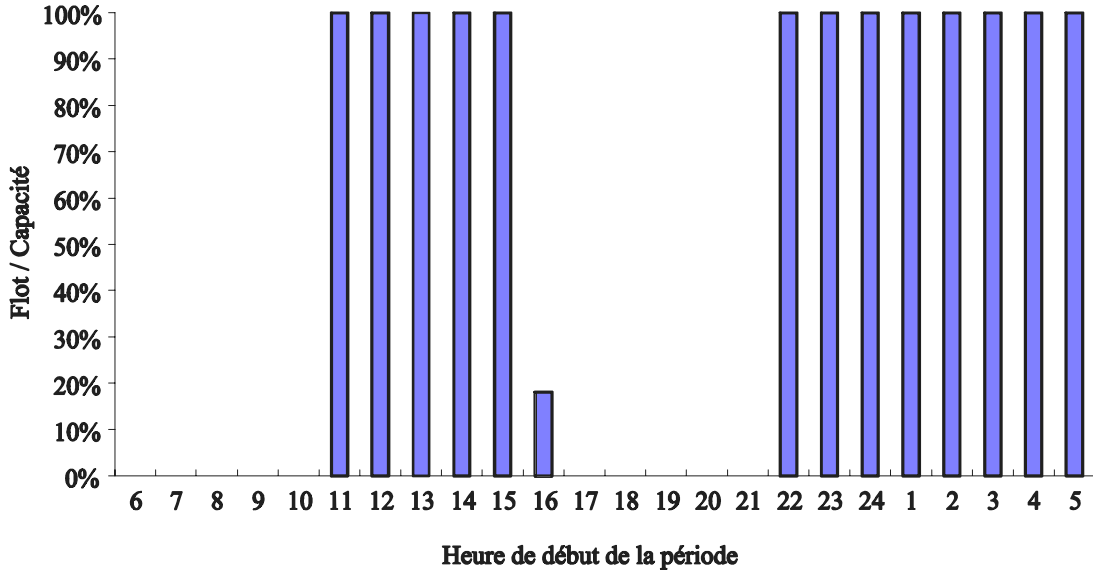


Figure 5.3 : Fonctionnement de l'usine respectant les contraintes de continuité

Pour la transformer en contraintes qui pourront alors s'ajouter à celles du programme linéaire, nous posons M réel strictement positif suffisamment grand (nous expliciterons ce que nous entendons par là plus loin) et nous introduisons une variable binaire supplémentaire y . On écrit alors :

$$f(x) \leq M(1 - y) \text{ et } g(x) \geq My$$

Nous obtenons bien l'implication recherchée puisque si $f(x) > 0, y = 0$ (la valeur 1 étant impossible) et donc $g(x) \geq 0$.

Ce que nous venons d'écrire montre qu'il faut prendre pour M un majorant de la fonction $f(x)$ de façon à ce que l'inégalité

$$f(x) \leq M \text{ ne soit pas mise en défaut lorsque } y = 0.$$

5.3.2.2 Implications de continuité

Traisons le cas où un des organes de commande du réseau est soumis à une contrainte de continuité. Notons k_0 l'indice de cet organe avec $k_0 \in \langle 1; k \rangle$. Notons L la durée minimale de fonctionnement de cet organe et A la durée minimale d'arrêt de ce même organe.

Nous avons besoin, par la suite, de définir une variable entière B comme

- $B = A - 1$, si A est entier,
- $B = \text{ent}(A)$, si A n'est pas entier.

Il faut traduire sous forme de contrainte le fait que si un $\alpha_i^g(\theta_0)$ est non nul et correspond à un vecteur commande faisant intervenir l'organe k_0 (c'est-à-dire un vecteur commande pour lequel la k_0^{eme} coordonnée vaut 1), la somme des $\alpha_i^g(\theta)$ des vecteurs de commande ayant cette composante égale à 1 devra valoir 1 et ce sur chacun des pas de temps suivants de façon à atteindre une durée minimale L .

On définit dans un premier temps $\Omega(i) = \left\{ i \in \langle 1; I \rangle / \left(\vec{K}_i \right)_{k_0} = 1 \right\}$.

On peut alors formuler les deux implications qui suivent :

$$(I_1) \left\{ \begin{array}{l} \text{Si } \exists \theta_0 \in \langle 2; T - L - 1 \rangle \text{ tel que } \left(\sum_{i \in \Omega(i)} \alpha_i^g(\theta_0) < 1 \right) \\ \text{et } \left(\sum_{i \in \Omega(i)} \alpha_i^g(\theta_0 - 1) = 1 \right) \\ \text{alors } \sum_{k=0}^{B+1} \sum_{i \in \Omega(i)} \alpha_i^g(\theta_0 + k) \leq B + 2 - A \end{array} \right. \quad (5.1)$$

$$(I_2) \left\{ \begin{array}{l} \text{Si } \exists \theta_0 \in \langle 2; T - L - 1 \rangle \text{ tel que } \left(\sum_{i \in \Omega(i)} \alpha_i^g(\theta_0) > 0 \right) \\ \text{et } \left(\sum_{k=0}^{B+1} \sum_{i \in \Omega(i)} \alpha_i^g(\theta_0 - k) \leq B + 2 - A \right) \\ \text{alors } \left(\sum_{k=1}^{L-1} \sum_{i \in \Omega(i)} \alpha_i^g(1 + k) = L - 1 \right) \\ \text{et } \left(\sum_{i \in \Omega(i)} (\alpha_i^g(1) + \alpha_i^g(1 + L)) \geq 1 \right) \end{array} \right. \quad (5.2)$$

Ces deux implications traduisent chacune un des aspects de la contrainte de continuité. L'équation (5.1) impose un arrêt minimal d'une durée de A à la fin d'une période de fonctionnement alors que l'équation (5.2) garantit un fonctionnement d'une durée minimale de L pour chaque plage de fonctionnement.

L'antécédent de chacune de ces deux implications n'est valable que pour

$\theta_0 \in \langle 2; T \rangle$ puisque les conditions portent à la fois sur $\alpha_i^g(\theta_0)$ et $\alpha_i^g(\theta_0 - 1)$. Aux deux implications précédentes il faut donc rajouter :

$$(I'_2) \left\{ \begin{array}{l} \text{Si } \left(\sum_{i \in \Omega(i)} \alpha_i^g(1) > 0 \right) \\ \text{alors } \left(\sum_{k=1}^{L-1} \sum_{i \in \Omega(i)} \alpha_i^g(1 + k) = L - 1 \right) \\ \text{et } \left(\sum_{i \in \Omega(i)} (\alpha_i^g(1) + \alpha_i^g(1 + L)) \geq 1 \right) \end{array} \right. \quad (5.3)$$

De façon symétrique, il faut traiter à part la fin de l'horizon d'optimisation. En effet, le conséquent de l'équation (5.2) fait intervenir les pas de temps de θ_0 à $\theta_0 + L$. Les écritures n'auront de sens que si $\theta_0 < T - L$.

Avant de modéliser par une implication, il est nécessaire d'arrêter une règle de gestion de cette fin d'horizon. La règle que nous choisissons peut s'exprimer ainsi : "Si l'organe est arrêté ou s'arrête au temps $(T - L)$, il, est maintenu à l'arrêt jusqu'au bout de l'horizon."

Cette règle s'écrit de la façon suivante :

$$(I_2'') \left\{ \begin{array}{l} Si \exists \theta \in \langle T - L; T - 1 \rangle \left(\sum_{i \in \Omega(i)} \alpha_i^g(\theta) < 1 \right) \\ \text{alors} \left(\sum_{k=1}^{T-\theta} \sum_{i \in \Omega(i)} \alpha_i^g(\theta + k) = 0 \right) \end{array} \right. \quad (5.4)$$

5.3.2.3 Variables intermédiaires

A chaque pas de temps θ , il faudra effectuer quatre tests :

- $\sum_{i \in \Omega(i)} \alpha_i^g(\theta) < 1$ ou $\sum_{i \in \Omega(i)} \alpha_i^g(\theta) = 1$,
- $\sum_{i \in \Omega(i)} \alpha_i^g(\theta - 1) < 1$ ou $\sum_{i \in \Omega(i)} \alpha_i^g(\theta - 1) = 1$,
- $\sum_{i \in \Omega(i)} \alpha_i^g(\theta) > 0$ ou $\sum_{i \in \Omega(i)} \alpha_i^g(\theta) = 0$,
- $\sum_{k=0}^{B+1} \sum_{i \in \Omega(i)} \alpha_i^g(\theta - k) \leq B + 2 - A$ ou $\sum_{k=0}^{B+1} \sum_{i \in \Omega(i)} \alpha_i^g(\theta - k) > B + 2 - A$.

Pour transcrire le résultat de ces tests nous introduisons quatre variables réelles, $p(\theta)$, $r(\theta)$, $s(\theta)$ et $t(\theta)$.

Les tests s'écriront alors :

- Si pour $\theta \in \langle 2; T - L - 1 \rangle$, $\sum_{i \in \Omega(i)} \alpha_i^g(\theta) < 1$ alors $p(\theta) \geq 2$ sinon $p(\theta) \leq 1$,
- Si pour $\theta_0 \in \langle 2; T - L - 1 \rangle$, $\sum_{i \in \Omega(i)} \alpha_i^g(\theta - 1) < 1$ alors $r(\theta) \leq 1$ sinon $r(\theta) \geq 2$,
- Si pour $\theta_0 \in \langle 2; T - L - 1 \rangle$, $\sum_{i \in \Omega(i)} \alpha_i^g(\theta) > 0$ alors $s(\theta) \geq 2$ sinon $s(\theta) \leq 1$,
- Si pour $\theta_0 \in \langle 2; T - L - 1 \rangle$, $\sum_{k=0}^{B+1} \sum_{i \in \Omega(i)} \alpha_i^g(\theta - k) > B + 2 - A$ alors $t(\theta) \leq 1$ sinon $t(\theta) \geq 2$.

Ces implications peuvent être transformées avec la méthode décrite au paragraphe 5.3.2.1 en utilisant 8 variables binaires par pas de temps ainsi que la constante $M > 2$:

$$\forall \theta \in \langle 2; T - L - 1 \rangle, \left\{ \begin{array}{l} \left\{ \begin{array}{l} 1 - \sum_{i \in \Omega(i)} \alpha_i^g(\theta) \leq M(1 - u_1(\theta)) \\ 2 - p(\theta) \leq Mu_1(\theta) \\ \sum_{i \in \Omega(i)} \alpha_i^g(\theta) + \varepsilon - 1 \leq M(1 - u_2(\theta)) \\ p(\theta) - 1 \leq Mu_2(\theta) \end{array} \right. \\ \left\{ \begin{array}{l} 1 - \sum_{i \in \Omega(i)} \alpha_i^g(\theta - 1) \leq M(1 - u_3(\theta)) \\ r(\theta) - 1 \leq Mu_3(\theta) \\ \sum_{i \in \Omega(i)} \alpha_i^g(\theta - 1) + \varepsilon - 1 \leq M(1 - u_4(\theta)) \\ 2 - r(\theta) \leq Mu_4(\theta) \end{array} \right. \\ \left\{ \begin{array}{l} \sum_{i \in \Omega(i)} \alpha_i^g(\theta) \leq M(1 - u_5(\theta)) \\ 2 - s(\theta) \leq Mu_5(\theta) \\ \varepsilon - \sum_{i \in \Omega(i)} \alpha_i^g(\theta) \leq M(1 - u_6(\theta)) \\ s(\theta) - 1 \leq Mu_6(\theta) \end{array} \right. \\ \left\{ \begin{array}{l} B + 2 - A + \varepsilon - \sum_{k=0}^{B+1} \sum_{i \in \Omega(i)} \alpha_i^g(\theta - k) \leq M(1 - u_7(\theta)) \\ 2 - t(\theta) \leq Mu_7(\theta) \\ \sum_{k=0}^{B+1} \sum_{i \in \Omega(i)} \alpha_i^g(\theta - k) - B - 2 + A \leq M(1 - u_8(\theta)) \\ t(\theta) - 1 \leq Mu_8(\theta) \end{array} \right. \end{array} \right. \quad (5.5)$$

5.3.2.4 Utilisation des variables intermédiaires

Maintenant que nous avons introduit les variables intermédiaires nécessaires, nous pouvons écrire les implications précédentes. Prenons, par exemple, l'équation (5.1) et remarquons que l'introduction de nouvelles variables nous permet d'écrire l'équivalence (5.6) à la condition de contraindre toutes les variables p , r , s et t à être strictement inférieures à 3 dans l'ensemble du problème. Ce majorant strict des variables intermédiaires nous permet, en effet, de mettre en place la réciproque de (5.6) qui n'était a priori

qu'une implication.

$$\begin{aligned} & \forall \theta \in \langle 2; T - L - 1 \rangle \left(\sum_{i \in \Omega(i)} \alpha_i^g(\theta_0) < 1 \right) \\ & \quad \text{et} \left(\sum_{i \in \Omega(i)} \alpha_i^g(\theta_0 - 1) = 1 \right) \\ & \iff \forall \theta \in \langle 2; T - L - 1 \rangle, p(\theta) + r(\theta) \geq 4 \end{aligned} \quad (5.6)$$

Alors l'équation (5.1) devient :

$$\forall \theta \in \langle 2; T - L - 1 \rangle, p(\theta) + r(\theta) \geq 4 \Rightarrow \sum_{k=0}^{B+1} \sum_{i \in \Omega(i)} \alpha_i^g(\theta + k) \leq B + 2 - A \quad (5.7)$$

Ce qui se traduit sous forme de contraintes par :

$$\forall \theta \in \langle 2; T - L - 1 \rangle \left\{ \begin{array}{l} 4 + \varepsilon - p(\theta) + r(\theta) \leq M(1 - u_9(\theta)) \\ \sum_{k=0}^{B+1} \sum_{i \in \Omega(i)} \alpha_i^g(\theta + k) - B - 2 + A \leq M u_9(\theta) \end{array} \right. \quad (5.8)$$

De la même façon l'équation (5.2) se traduit par :

$$\begin{aligned} & \forall \theta \in \langle 2; T - L - 1 \rangle s(\theta) + t(\theta) \geq 4 \\ & \Rightarrow \left(\sum_{k=1}^{L-1} \sum_{i \in \Omega(i)} \alpha_i^g(\theta + k) = L - 1 \right) \\ & \quad \text{et} \left(\sum_{i \in \Omega(i)} (\alpha_i^g(\theta) + \alpha_i^g(\theta + L)) \geq 1 \right) \end{aligned} \quad (5.9)$$

Ce qui s'écrit sous forme de contraintes par :

$$\forall \theta \in \langle 2; T - L - 1 \rangle \left\{ \begin{array}{l} 4 - \varepsilon - s(\theta) - t(\theta) \leq M(1 - u_{10}(\theta)) \\ L - 1 - \sum_{k=1}^{L-1} \sum_{i \in \Omega(i)} \alpha_i^g(\theta + k) \leq M u_{10}(\theta) \\ 1 - \sum_{i \in \Omega(i)} (\alpha_i^g(\theta) + \alpha_i^g(\theta + L)) \leq M u_{10}(\theta) \end{array} \right. \quad (5.10)$$

5.3.2.5 Variables intermédiaires dans la fonction objectif

Comme nous l'avons vu au paragraphe 5.1, nous cherchons parfois à minimiser le nombre de changement d'état des organes présents sur le réseau. Dans ce cas nous ne ferons que calculer les variables intermédiaires sans utiliser celles-ci dans un deuxième temps. Le modèle obtenu sera plus simple de par la disparition de cette deuxième étape. Nous ne reviendrons pas par la suite sur l'écriture de ce modèle.

5.3.3 Exemple d'utilisation de la méthode

5.3.3.1 Le réseau exemple

Le réseau utilisé est celui de la figure 5.4.

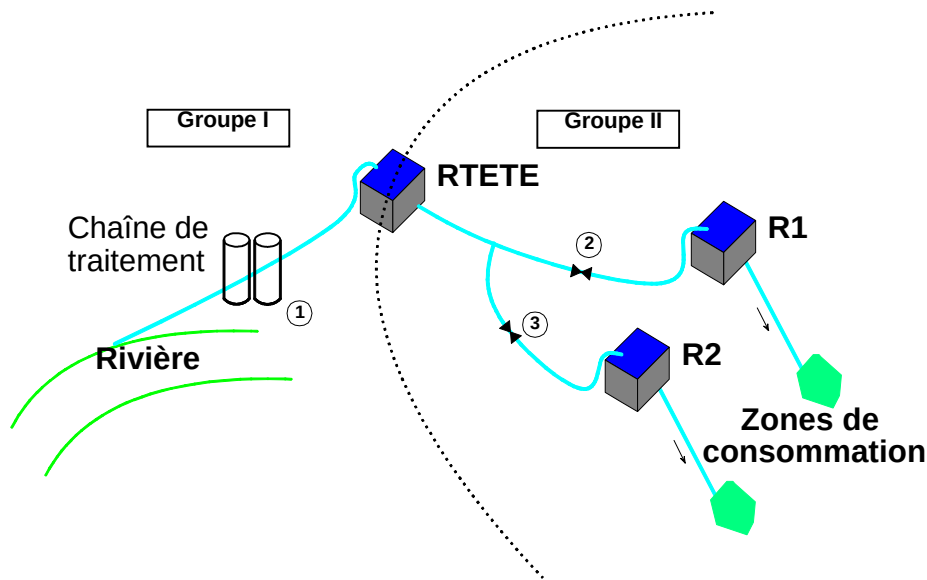


Figure 5.4 : Illustration de la contrainte de continuité sur un exemple

5.3.3.2 Les organes de commande

Ce réseau comporte 3 organes de commande : deux vannes d'entrée dans les réservoirs (numérotées 2 et 3) et une chaîne de traitement (numérotée 1). Cette dernière comprend à la fois la mise en oeuvre de réactifs et le pompage de l'eau traitée mais sera considérée comme une seule commande.

La présence du réservoir de tête RTETE permet le partage en deux groupes de commande. Nous aurons donc 6 inconnues générales dans notre problème que nous noterons de la façon suivante :

- $\forall \theta \in \langle 1; 24 \rangle, \alpha_1^1(\theta)$ et $\alpha_2^1(\theta)$ pour le fonctionnement des organes du groupe I,
- $\forall \theta \in \langle 1; 24 \rangle, \alpha_1^2(\theta), \alpha_2^2(\theta), \alpha_3^2(\theta)$ et $\alpha_4^2(\theta)$ pour le fonctionnement des organes du groupe II.

Ces variables vérifient : $\forall \theta \in \langle 1; 24 \rangle, \forall g \in \langle 1; 2 \rangle, \forall i \in \langle 1; I_g \rangle, 0 \leq \alpha_i^g(\theta) \leq 1$

Les différents vecteurs de commande s'écrivent pour leur part :

$$\begin{aligned} \text{pour } 1 : \vec{K}_1^1 &= |0 \quad \vec{K}_2^1 = |1 \\ \text{pour } 2 \text{ et } 3 : \vec{K}_1^2 &= \begin{vmatrix} 0 \\ 0 \end{vmatrix} \quad \vec{K}_2^2 = \begin{vmatrix} 1 \\ 0 \end{vmatrix} \quad \vec{K}_3^2 = \begin{vmatrix} 0 \\ 1 \end{vmatrix} \quad \vec{K}_4^2 = \begin{vmatrix} 1 \\ 1 \end{vmatrix} \end{aligned}$$

5.3.3.3 Données simulées

Nous avons fixé les valeurs de consommation heure par heure sur chacun des réservoirs. Pour obtenir les débits entrant dans les réservoirs, nous avons procédé à une modélisation numérique. Compte tenu des caractéristiques des canalisations, le débit entrant dans le réservoir R1 est de 30 l/s lorsque sa vanne est ouverte et de 38 l/s pour R2. La capacité de production de la chaîne de traitement est de 138 l/s. Nous aurons donc les valeurs suivantes des débits : $\forall \theta \in \langle 1; 24 \rangle, Q_{1HEAD}^1(\theta) = 0; Q_{2HEAD}^1(\theta) = 138; Q_{1HEAD}^2(\theta) = 0; Q_{2HEAD}^2(\theta) = -30; Q_{3HEAD}^2(\theta) = -38; Q_{4HEAD}^2(\theta) = -68$

5.3.3.4 Ecriture du modèle

Le modèle prend la forme de l'équation (2.9) avec, en particulier, $T = 24, R = 3$ et $G = 2$. Ce modèle s'écrit alors sous la forme (5.11).

$$\left\{ \begin{aligned} z_{\min} &= \sum_{\theta=1}^{24} \sum_{g=1}^2 \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) \\ \forall \theta \in \langle 1; 23 \rangle, \forall r \in \langle 1; 3 \rangle, V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^2 \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^{\theta} C_r(t) &\leq V_{r\max} \\ \forall \theta \in \langle 1; 23 \rangle, \forall r \in \langle 1; 3 \rangle, V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^2 \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^{\theta} C_r(t) &\geq V_{r\min} \\ \forall r \in \langle 1; 3 \rangle, V_r(0) + \sum_{t=1}^{24} \sum_{g=1}^2 \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^{24} C_r(t) &= V_{r\max} \\ \forall \theta \in \langle 1; 24 \rangle, \forall g \in \langle 1; 2 \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) &= 1 \\ \forall \theta \in \langle 1; 24 \rangle, \forall g \in \langle 1; 2 \rangle, \forall i \in \langle 1; I_g \rangle, \alpha_i^g(\theta) &\geq 0 \end{aligned} \right. \quad (5.11)$$

Le seul transit ayant un coût est celui représentant la chaîne de traitement (les autres transits étant gravitaires, ils n'entraînent pas de coûts proportionnels). Les différents coûts unitaires vérifieront donc : $\forall \theta \in \langle 1; 24 \rangle, \pi_1^1(\theta) = \pi_1^2(\theta) = \pi_2^2(\theta) = \pi_3^2(\theta) = \pi_4^2(\theta) = 0$ et $\pi_2^1(\theta) \neq 0$.

5.3.3.5 Contrainte de continuité

5.3.3.5.1 Organe concerné La chaîne de traitement peut être soumise à une contrainte de continuité. Nous poserons qu'elle ne peut fonctionner que par plages continues de 4

heures minimum avec des périodes d'arrêt d'au moins 1 heure. Seule la variable $\alpha_2^1(\theta)$ sera concernée par la contrainte de continuité.

5.3.3.5.2 Contraintes en cours d'horizon Les différentes contraintes à rajouter se présenteront de la façon suivante :

$$\forall \theta \in \langle 1; 19 \rangle, \left\{ \begin{array}{l} 1 - \alpha_2^1(\theta) \leq M(1 - u_1(\theta)) \\ 2 - p(\theta) \leq Mu_1(\theta) \\ r(\theta - 1) - 1 \leq Mu_1(\theta) \\ \alpha_2^1(\theta) + \varepsilon - \leq M(1 - u_2(\theta)) \\ p(\theta) - 1 \leq Mu_2(\theta) \\ 2 - r(\theta - 1) \leq Mu_2(\theta) \end{array} \right. \quad (5.12)$$

$$\forall \theta \in \langle 2; 19 \rangle, \left\{ \begin{array}{l} \left\{ \begin{array}{l} \alpha_2^1(\theta) \leq M(1 - u_3(\theta)) \\ 2 - s(\theta) \leq Mu_3(\theta) \\ \varepsilon - \alpha_2^1(\theta) \leq M(1 - u_4(\theta)) \\ s(\theta) - 1 \leq Mu_4(\theta) \end{array} \right. \\ \left\{ \begin{array}{l} 1 + \varepsilon - \sum_{k=0}^1 \alpha_2^1(\theta - k) \leq M(1 - u_5(\theta)) \\ 2 - t(\theta) \leq Mu_5(\theta) \\ \sum_{k=0}^1 \alpha_2^1(\theta - k) - 1 \leq M(1 - u_6(\theta)) \\ t(\theta) - 1 \leq Mu_6(\theta) \end{array} \right. \\ \left\{ \begin{array}{l} \varepsilon + p(\theta) + r(\theta) - 4 \leq M(1 - u_7(\theta)) \\ \sum_{k=0}^1 \alpha_2^1(\theta + k) - 1 \leq Mu_7(\theta) \end{array} \right. \\ \left\{ \begin{array}{l} \varepsilon + s(\theta) + t(\theta) - 4 \leq M(1 - u_8(\theta)) \\ 3 - \sum_{k=1}^3 \alpha_2^1(\theta + k) \leq Mu_8(\theta) \\ 1 - \alpha_2^1(\theta) - \alpha_2^1(\theta + 4) \leq Mu_8(\theta) \end{array} \right. \end{array} \right. \quad (5.13)$$

Un retour rapide sur le modèle nous permet de rappeler que les variables p , r , s et t calculées grâce à (5.12) et aux deux premières contraintes de (5.13) sont des variables intermédiaires qui permettent au modèle de gérer les arrêts et les redémarrages de l'organe. Ainsi le calcul de ces variables intermédiaires permet-il d'imposer :

- par la troisième contrainte de (5.13) un arrêt minimal de 1 heure. En effet, cette contrainte traduit la limitation à une heure de la somme de deux valeurs consécutives de $\alpha_2^1(\theta)$ lorsque les conditions d'arrêt ont été vérifiées.

- Par la quatrième contrainte de (5.13) un fonctionnement minimal de 4 heures lorsque les conditions de redémarrage ont été vérifiées.

5.3.3.5.3 Contraintes de début et de fin d’horizon Il est également nécessaire d’ajouter une série de contraintes pour le premier pas de temps de l’horizon ainsi que pour les quatre derniers :

$$\begin{cases} \alpha_2^1(1) \leq M(1 - u_9(1)) \\ 3 - \sum_{k=1}^3 \alpha_2^1(1+k) \leq Mu_9(1) \\ 1 - \alpha_2^1(1) - \alpha_2^1(5) \leq Mu_9(1) \end{cases} \quad (5.14)$$

$$\forall \theta \in \langle 20; 23 \rangle \begin{cases} 1 - \alpha_2^1(\theta) \leq M(1 - u_{10}(\theta)) \\ \sum_{k=1}^{T-\theta} \alpha_2^1(\theta+k) \leq Mu_{10}(\theta) \end{cases} \quad (5.15)$$

5.3.3.6 Taille du problème

Pour mesurer le surcoût en taille de programme et donc en temps de calcul, nous allons quantifier le nombre de variables et de contraintes que notre programme comporte.

Nous allons pour cela utiliser trois constantes :

- T : nombre de pas de temps, valant ici 24,
- R : nombre de réservoirs valant ici 3,
- I' : nombre de vecteurs de commande indépendants que l’on peut écrire $I' = \sum_{g=1}^G I_g$ valant ici $2 + 4 = 6$.

5.3.3.6.1 Variables réelles Nous ne tenons pas compte ici des variables d’écart nécessaires à la résolution elle-même. Avant la mise en place des variables permettant de contraindre l’usine à fonctionner de façon continue, les seules variables réelles sont les $\alpha_i^g(\theta)$, on en dénombre $T \times I'$, soit 124. L’ajout de la continuité fait intervenir 19×2 (p et r) et 18×2 (s et t) variables supplémentaires comme cela apparaît dans les équations (5.12) et (5.13), soit 74 variables réelles supplémentaires.

5.3.3.6.2 Variables booléennes Les variables booléennes apparaissent numérotées de u_1 à u_{10} . Leur nombre respectif dépend de la partie de l’horizon sur laquelle elles sont utilisées. Il y a 19 variables u_1 et u_2 , 18 variables u_3 à u_8 , 1 variable u_9 et 4 variables u_{10} . En tout, cela représente 151 variables binaires.

5.3.3.6.3 Contraintes Dans le programme général (5.11), nous pouvons déjà recenser dans le tableau 5.2 les contraintes avant d'intégrer celles propres au fonctionnement continu de l'usine.

Type de contraintes	Nombre
Volume max	$R \times (T - 1) = 69$
Volume min	$R \times (T - 1) = 69$
Volume final	$R = 3$
Somme des $\alpha_i^g(\theta)$	$T \times G = 48$
TOTAL	189

Tableau 5.2 : Nombre de contraintes dans le modèle sans contraintes de continuité

Dans ce tableau nous ne tenons pas compte des contraintes de positivité. En se reportant aux équations (5.12), (5.13), (5.14) et (5.15) nous pouvons dénombrer 359 contraintes ajoutées pour tenir compte de la continuité de fonctionnement.

5.3.4 Résolution pratique

5.3.4.1 Conditions de résolution

Nous utiliserons le solveur en nombre mixtes de Cplex appelé depuis des programmes construits en langage C. L'algorithme de résolution utilisé est un algorithme de "Branch and Bound" (séparation-évaluation). L'idée de cet algorithme est de s'affranchir d'un examen de toutes les solutions (qui représenterait ici 2^{151} possibilités) tout en s'assurant d'obtenir la solution optimale. On construit donc un arbre pour lequel le niveau zéro correspond à l'ensemble des solutions possibles et chaque niveau inférieur contraint une variable supplémentaire. Voyons, figure 5.5, la construction de cet arbre sur un exemple à trois variables binaires tiré de (Minoux, 1983b).

Sur le dessin de la figure 5.5 le sommet S représente l'ensemble des $2^3 = 8$ solutions du problème. Une fois un tel arbre construit, il est nécessaire de trouver à chaque sommet une évaluation par défaut de la meilleure solution grâce à une fonction d'évaluation f . Cette meilleure évaluation s'écrira $f(S_i) \leq \min_{x \in S_i} \{c.x\}$. L'intérêt de cette évaluation est que si l'on connaît une solution du problème en nombre entier de coût $\bar{z} = c \bar{x}$, obtenue au cours des étapes précédentes de parcours de l'arbre ou par une méthode approchée, et si $f(S_i) > \bar{z}$ alors on saura qu'aucune solution contenue dans S_i ne peut être optimale.

Pour que l'algorithme trouve rapidement une solution optimale, il est donc nécessaire de trouver des solutions entières proches de l'optimum final et si possible le plus haut

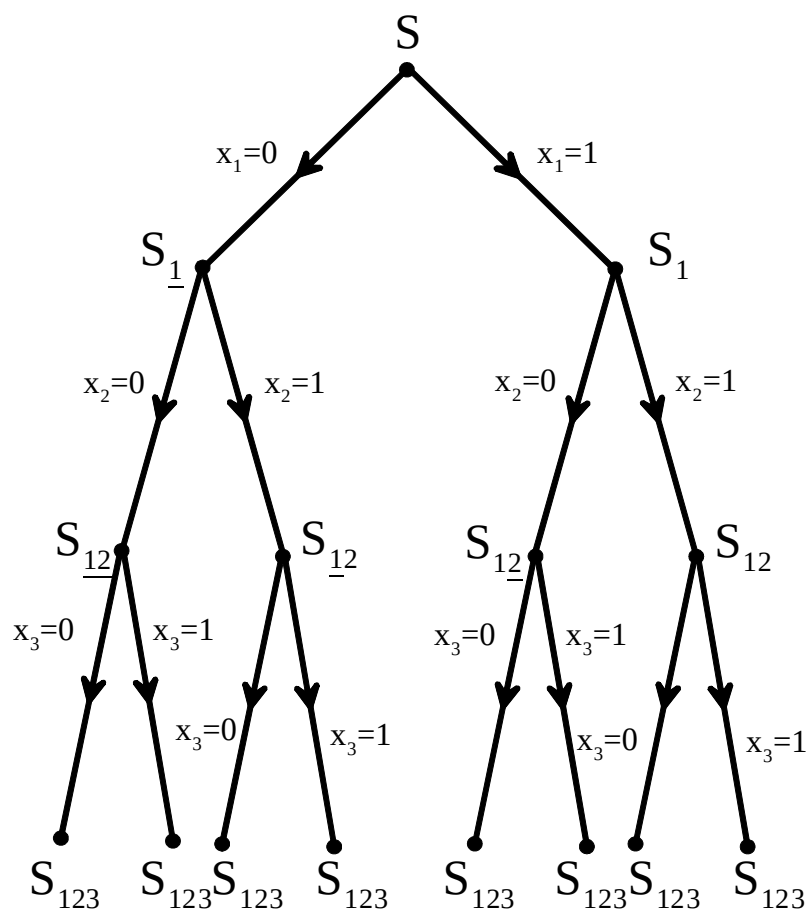


Figure 5.5: Arborescence d'un problème en nombres entiers à 3 variables binaires (Minoux, 1983b)

possible dans l'arbre. Nous verrons un peu plus loin les paramètres que nous pouvons utiliser pour tenter de concilier ces deux objectifs. Dans un premier temps, nous donnons des résultats obtenus en laissant le solveur choisir au mieux l'ordre de parcours des noeuds de l'arbre.

Pour évaluer la complexité des calculs supplémentaires, nous pourrions utiliser l'un ou plusieurs des critères suivants :

- le nombre de noeuds parcourus avant l'arrêt de l'algorithme,
- le nombre d'itérations de l'algorithme,
- le temps de résolution mis par un Pentium II 300Mhz.

Enfin, il est bon de rappeler que si les conditions de déroulement de l'algorithme imposaient de garder la quasi-totalité de l'arbre il s'agirait de parcourir $2^{151} = 3.10^{45}$

qui même avec une vitesse de parcours de 300 noeuds à la seconde représenterait 3.10^{35} années de calcul.

5.3.4.2 Utilisation de fonctions coût artificielles

5.3.4.2.1 Définition des fonctions utilisées Pour comparer les temps de calcul et pouvoir ainsi visualiser le temps supplémentaire dû à l'introduction des contraintes de continuité, nous avons testé deux fonctions coûts différentes pour π_2^1 . Ces deux fonctions sont représentées sur le graphique 5.6.

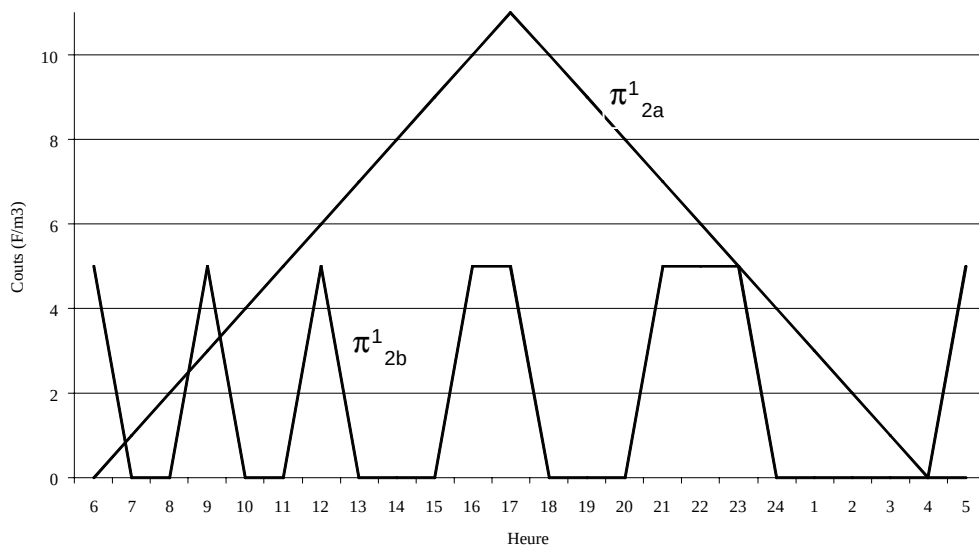
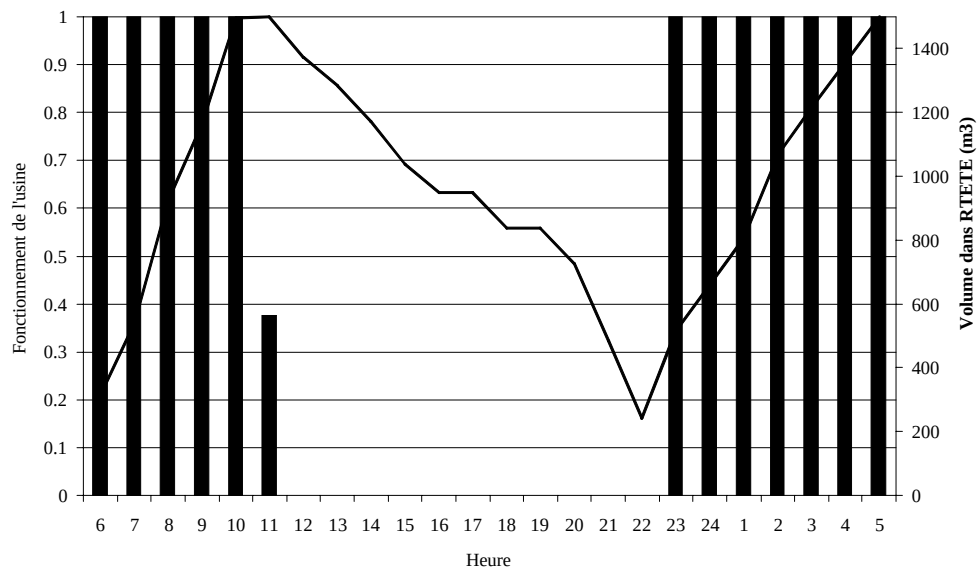


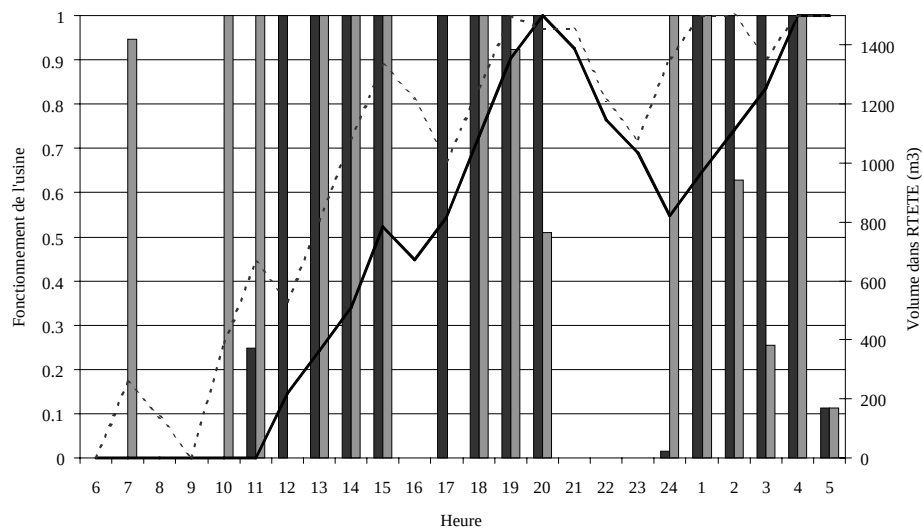
Figure 5.6 : Forme des fonctions coût de validation de la méthode

5.3.4.2.2 Respect de la contrainte de continuité La fonction de coût $\pi_{2a}^1(t)$ a été choisie de façon à favoriser l'obtention d'une solution continue sans avoir à l'imposer dans le modèle. A l'inverse la fonction $\pi_{2b}^1(t)$ devrait fournir une solution ne respectant pas les contraintes de continuité si celles-ci ne sont pas imposées.

Les résultats obtenus sont conformes à ce qui pouvait être imaginé a priori. Lorsque le coût $\pi_{2a}^1(t)$ est utilisé, les valeurs de $\alpha_2^1(\theta)$ sont les mêmes pour le modèle respectant la contrainte de continuité et pour celui qui ne la respecte pas. Nous obtenons toujours la représentation de la figure 5.7 où la valeur de la commande $\alpha_1^2(\theta)$ est représentée par des barres comprises entre 0 et 1 ; le trait plein représente, pour sa part, le niveau d'eau dans le réservoir RTETE.

Figure 5.7 : Résultat avec l'utilisation de la fonction coût $\pi_{2a}^1(t)$

A l'opposé, nous pouvons voir sur le graphique de la figure 5.8 l'efficacité de la méthode permettant de tenir compte des contraintes de continuité.

Figure 5.8 : Résultat avec l'utilisation de la fonction coût $\pi_{2b}^1(t)$

Sur le graphique 5.8, les barres représentent toujours la valeur de commande de l'usine de traitement de l'eau et le trait plein le niveau d'eau dans le réservoir RTETE. Les

résultats du modèle où l'on n'impose pas de contraintes de continuité de fonctionnement de l'usine sont représentés par des barres claires et un trait pointillé. La méthode donne donc les résultats escomptés puisque l'on passe d'un résultat comportant des plages de fonctionnement inférieures à 4 heures au début de l'horizon d'optimisation (barres claires) à un fonctionnement respectant les contraintes de continuité (barres sombres).

5.3.4.2.3 Comparaison des modèles Il nous importe de regarder ce que l'intégration des contraintes de continuité sur l'organe $n^{\circ} 1$ ajoute comme complexité dans notre calcul. Le dénombrement des variables et des contraintes de notre modèle figure dans le tableau 5.3.

	Modèle sans contrainte de continuité	Modèle avec contraintes de continuité
Nombre de contraintes	189	544
Nombre de variables réelles	216	288
Nombre de variables binaires		151

Tableau 5.3 : Comparaison du nombre de variables dans chaque modèle

Avant d'exécuter le calcul lui-même, le solveur Cplex effectue un calcul préliminaire ou "presolve" qui permet de réduire la taille des problèmes traités. Voyons, par exemple dans le tableau 5.4 la nouvelle dimension du problème après presolve dans le cas de la première fonction coût $\pi_{2a}^1(t)$.

	Modèle sans contrainte de continuité	Modèle avec contraintes de continuité
Variables	187	412
Contraintes	96	451

Tableau 5.4 : Nombre de variables et de contraintes après presolve

Le respect de la contrainte de continuité peut donc être très gourmand en temps de calcul si l'on part d'une situation où le fonctionnement de l'usine ne respecte initialement la contrainte de continuité. Plus précisément, si nous examinons le cheminement dans l'arborescence au cours de la résolution de l'algorithme, nous observons qu'après quelques

		Modèle sans contrainte de continuité	Modèle avec contraintes de continuité
Fonction coût $\pi_{2a}^1(t)$	Temps de calcul (s)	1	9
	Nombre total d'itérations	121	2546
	Fonction objectif	1	1
Fonction coût $\pi_{2b}^1(t)$	Temps de calcul (s)	1	5770
	Nombre total d'itérations	160	8403221
	Fonction objectif	1	19,2

Tableau 5.5 : Calculs supplémentaires dus à la contrainte de continuité

itérations (environ 20 000 qui correspond à 8 secondes) une première solution entière est mise en évidence avec une valeur de la fonction objectif ne s'écartant que de 9 % de la solution optimale finale. Enfin, et c'est l'élément déterminant ici, une solution entière de même coût que le problème relaxé en variables réelles est trouvé après 34 973 itérations. Le calcul est alors stoppé puisqu'il est alors impossible de trouver une solution de coût plus faible.

Dans le cas de la fonction coût $\pi_{2b}^1(t)$ il n'existe pas de solution entière de même coût. C'est pour cette raison que les temps de calcul sont aussi importants. Ceux-ci sont néanmoins réduits par la mise en évidence d'une solution entière après seulement 2 322 itérations d'une solution s'écartant de la solution finalement retenue de 12,5 %. La suite des calculs est donc raccourci par l'utilisation de ce majorant comme nous l'avons précisé au paragraphe 5.3.4.1. La solution retenue finalement est exhibée après 8 245 388 itérations, le reste du calcul étant consacré à vérifier qu'il n'existe plus dans l'arborescence de solutions relaxées meilleures que la meilleure solution entière.

5.3.4.3 Utilisation d'une fonction coût réelle

5.3.4.3.1 Définition de la fonction utilisée Les fonctions coûts que nous avons utilisées précédemment ont évidemment été bâties spécialement pour la démonstration précédente. Il est également intéressant de regarder les deux modèles lorsque nous utilisons une fonction coût plus proche de la réalité. Ainsi, nous avons choisi de travailler avec trois plages de coût sur la journée : les heures creuses, les heures pleines et les heures de pointe. La forme de la fonction coût sur la journée peut alors être vue sur la figure 5.9.

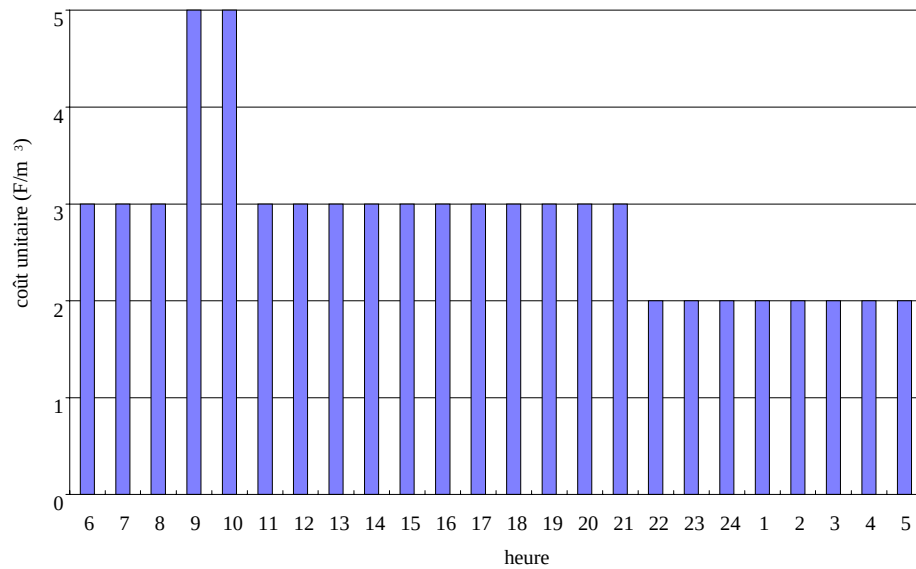


Figure 5.9 : Forme de la fonction coût utilisant les tarifs EDF

5.3.4.3.2 Comparaison des modèles Nous obtenons un fonctionnement de la chaîne de traitement qui ne respecte pas initialement la contrainte de continuité. Voyons ces résultats sur la figure 5.10.

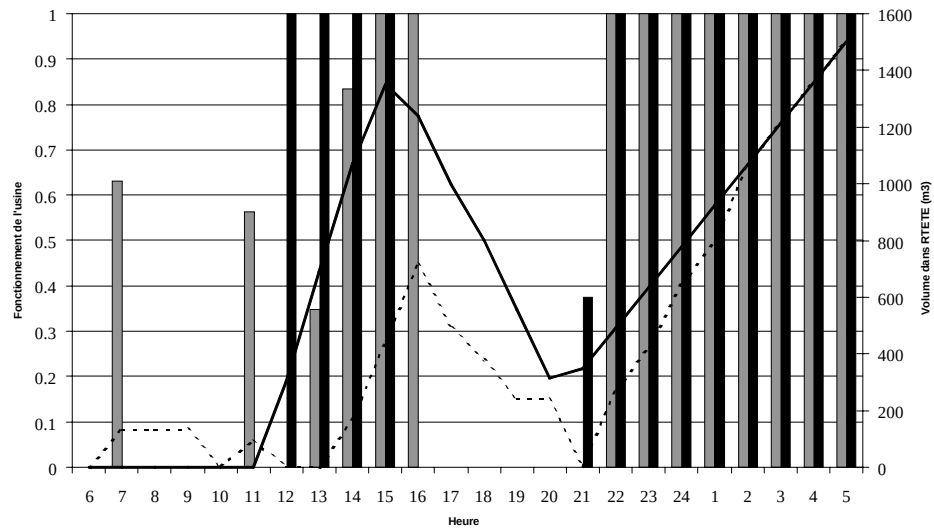


Figure 5.10 : Résultat avec l'utilisation des coûts réels

Sur la figure 5.10, nous représentons toujours en barres claires et en trait pointillé le

résultat du modèle qui n'impose pas le respect de la contrainte de continuité.

Voyons tableau 5.6 les temps de calcul nécessaires.

	Modèle sans contrainte de continuité	Modèle avec contraintes de continuité
temps de calcul (s)	1	15
Nombre total d'itérations	154	34973
Fonction objectif	1	1

Tableau 5.6 : Temps de calcul avec les coûts reels

Comme ce que nous avons observé avec la fonction $\pi_{2a}^1(t)$, l'algorithme avec cette fonction coût permet d'exhiber rapidement plusieurs solutions entières dont la dernière est de coût identique à la solution relaxée ce qui arrête les calculs.

5.3.4.4 Choix des paramètres de l'algorithme

A plusieurs reprises nous avons observé des temps de calcul réduits par la mise en évidence au cours des calculs de solutions entières admissibles réduisant la partie de l'arborescence à explorer. Il est important de discuter du choix des paramètres de déroulement de cet algorithme de façon à pouvoir éventuellement réduire ces temps de calcul et surtout de réussir à les maintenir proches de ce qu'ils sont, même en cas de modification des conditions. Nous allons étudier trois types de paramètres : les options de parcours de l'arborescence, le choix d'une solution entière initiale et la tolérance de sous-optimalité acceptée.

5.3.4.4.1 Paramètres de parcours de l'arborescence Plusieurs paramètres permettent de choisir le parcours sur l'arborescence. Les deux principaux sont le sens de parcours de l'arbre et le choix de l'ordre des variables astreintes à être entières. La première des deux options consiste à choisir si lorsque l'on se trouve à un noeud, on choisit plutôt de remonter vers le niveau 0 de l'arborescence, de descendre le plus bas possible ou de laisser l'algorithme choisir à chaque noeud.

Ces deux options ont été essayées sur les différents modèles que nous avons bâti. Malheureusement le résultat n'est pas homogène puisque dans certain cas il est intéressant, par exemple, de rapidement descendre l'arborescence pour trouver une solution entière admissible qui joue bien le rôle de majorant du problème. Dans d'autres cas il est plus intéressant dans des niveaux faibles pour supprimer une partie plus importante

de l'arbre lorsque qu'une solution entière admissible est mise en évidence. Il n'est pas possible a priori de savoir dans quel cas nous nous situons sans avoir résolu le problème.

5.3.4.4.2 Choix d'une solution initiale Il est possible d'introduire un fichier donnant les valeurs initiales que l'on souhaite voir prendre par les variables astreintes à être entières. Le solveur débute les calculs en essayant ce jeu de données. Cet essai peut se traduire par trois résultats différents :

1. La solution proposée est infaisable. Ceci signifie qu'avec ces valeurs pour les variables entières, plusieurs contraintes du problème sont incompatibles entre elles.
2. La solution proposée donne bien une solution et, de plus, le coût de cette solution est le même que celui du problème identique avec toutes les variables entières relaxées. Le calcul est donc immédiatement terminé.
3. La solution proposée est faisable mais rien ne permet de conclure à l'obtention d'une solution optimale. Le parcours de l'arbre se fait donc à partir de cette solution initiale.

Dans le premier cas, le parcours de l'arbre sera strictement identique à ce qu'il serait sans utilisation d'une solution initiale. le dernier cas n'est pas forcément synonyme de gain de temps puisqu'il est possible que le point de départ de l'arborescence amène à perdre du temps à trouver une deuxième solution entière admissible plus intéressante. Nous avons rencontré cette difficulté lorsque nous avons testé la méthode avec la fonction coût $\pi_{2b}^1(t)$. Après la calcul d'optimisation initial, nous avons relancé un deuxième calcul en utilisant les solutions du premier comme solutions initiales au second. Dans ce cas nous gagnons bien du temps à trouver la première solution entière admissible (aucune itération au lieu des 2 322 nécessaires précédemment) par contre nous ne trouvons plus d'autres solutions entières admissibles et le calcul rencontre des problèmes de stockage mémoire après 12 heures de calcul sans avoir pu trouver une solution optimale.

Nous avons opéré la même simulation avec les coûts réels en utilisant en solutions initiales les résultats d'un premier calcul. Nous sommes bien dans le deuxième cas et l'algorithme s'arrête immédiatement puisque la solution est admissible et de coût identique à celui du problème relaxé.

Dans un deuxième temps, nous avons simulé une situation beaucoup plus intéressante d'un point de vue pratique. Il s'agit, en effet, d'utiliser les résultats d'un calcul pour initialiser l'optimisation d'un modèle différent. L'application de cette méthode répond directement à notre problème général puisque nous pourrions envisager d'optimiser le

fonctionnement d'un réseau d'eau potable sur 24 heures en utilisant les résultats de la veille.

Nous avons donc repris le modèle utilisé précédemment en multipliant la consommation par un coefficient de 1,2. La simulation donne un résultat intéressant. La solution est, en effet différente d'un point de vue temps de fonctionnement total de l'usine de production d'eau potable et d'un point de vue coût de cette solution puisqu'il s'agit de répondre à une demande plus forte. Pourtant le jeu de valeurs initiales pour les variables astreintes à être entières peut rester le même. Nous nous retrouvons dans le cadre des situations de type 2. La solution proposée initialement fournit une solution entière admissible dont la valeur de la fonction objectif est égale à celle obtenue en relaxant toutes les variables entières. Les calculs s'arrêtent donc immédiatement.

5.3.4.4.3 Tolérance de sous-optimalité Si nous reproduisons la simulation du paragraphe précédent en multipliant cette fois la demande horaire de chaque réservoir par 1,5, nous nous retrouvons dans ce que nous avons appelé une situation de type 3. Les calculs se poursuivent donc pour déterminer une solution entière de coût plus faible. Nous retrouvons la même difficulté que celle décrite lors de l'utilisation de la fonction coût $\pi_{2b}^1(t)$: les calculs durent plusieurs heures et finissent par saturer la mémoire de l'ordinateur utilisé. Nous avons donc choisi de modifier la valeur d'un paramètre valant par défaut 10^{-4} et nommé "tolerance mipgap". Ce coefficient que l'on notera MIPGAP peut être défini comme le rapport suivant :

$$\text{MIPGAP} = \frac{|\text{meilleure fonction objectif du problème relaxé} - \text{meilleure fonction objectif du problème entier}|}{1 + |\text{meilleure fonction objectif du problème relaxé}|}$$

Avec MIPGAP=0,05, nous obtenons bien la situation voulue ; c'est-à-dire un arrêt des calculs après la mise en évidence de la première solution entière admissible qui représente une perte d'optimalité de 4 % par rapport au problème relaxé sans que l'on puisse déterminer l'écart par rapport à l'optimum en variables entières.

5.4 Conclusion

La contrainte de continuité sur certains organes d'un réseau d'eau potable est donc bien un cas où la programmation par les graphes semble ne pas apporter de réponse satisfaisante. La possibilité d'introduire des variables binaires dans un programme linéaire permet de bien modéliser ce problème. L'introduction de ces nouvelles variables peut dans certains cas de coûts artificiels impliquer des temps de calcul incompatibles avec une gestion en temps réel. Dans le cas de coûts réels, il est possible d'initialiser les

calculs à partir de la solution de la veille. De plus, nous pourrons choisir d'accepter de perdre en optimalité pour s'assurer de trouver une solution satisfaisant la contrainte de continuité tout en maîtrisant les temps de calcul.

Enfin, il reste possible d'utiliser une partie de la méthode proposée pour minimiser le nombre de démarrage d'une pompe ou d'ouverture d'une vanne sans pour cela aboutir à des temps de calcul importants.

Chapitre 6

Choix entre plusieurs solutions de même coût

6.1 Le problème général

Le problème d'optimisation que nous avons proposé jusqu'à maintenant qu'il soit modélisé par les graphes ou par la programmation linéaire laisse apparaître des propriétés de régularité du fait de la forme simple de la fonction coût (présentée par exemple figure 5.9).

Si l'on cherche uniquement à minimiser les coûts proportionnels de production et de transport de l'eau sur le réseau, le calcul d'optimisation peut conduire à une solution non unique. Les solutions multiples obtenues diffèrent par des choix de pompages au cours d'une même plage de coût électrique. On peut notamment observer cela dans le chapitre 5 et en particulier dans les tableaux 5.1 et 5.5 où l'on voit que l'on peut, dans certains cas, imposer une contrainte de continuité sans modifier le coût de la solution optimale.

L'idée qui vient alors est d'enrichir le modèle d'optimisation en complétant le critère à optimiser. En effet, il est possible de hiérarchiser les solutions de même coût au sens du critère initial. Cette hiérarchisation peut se faire selon au moins deux axes différents :

- la continuité de fonctionnement d'une partie ou de l'ensemble des organes de commande du réseau. L'objectif d'une telle hiérarchisation se rapproche de celui du chapitre 5. La différence entre les deux approches est qu'ici la contrainte de continuité ne serait pas forcément vue comme une obligation mais comme un objectif guide.
- Le niveau dans des réservoirs à certains pas de temps. Cela revient à ajouter un

critère sur l'état du réseau alors que le critère précédent ne portait que sur les commandes.

6.2 Méthodes de résolution

6.2.1 Programmation multi-objectif

Cette méthode classique de la programmation mathématique consiste, dans un même programme, à combiner plusieurs critères. Elle semble donc a priori être adaptée à notre problème.

La méthode peut s'appliquer à plusieurs outils de la programmation mathématique comme la programmation par les graphes ou la programmation linéaire. Nous allons présenter la programmation linéaire multi-objectifs.

6.2.1.1 Choix de la fonction objectif

Parmi les deux possibilités évoquées au paragraphe 6.1 nous retenons comme deuxième objectif le fait d'optimiser les volumes des réservoirs. Plus précisément, nous voulons allier la minimisation des coûts de production et de transport de l'eau dans le réseau avec la volonté de maximiser les volumes dans les réservoirs. Cela nous permet, à coût équivalent, de retenir une solution plus sécuritaire.

$$z_{\min} = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) + \sum_{\theta=1}^T \sum_{r=1}^R \beta_{r\theta} (V_{r\max} - V_r(\theta)) \quad (6.1)$$

Les coefficients $\beta_{r\theta}$ permettent de donner un poids variable aux différents volumes en fonction de l'heure et du réservoir considéré. On peut, par exemple, privilégier un réservoir desservant une zone sensible aux heures où la consommation prévue est la plus importante.

6.2.1.2 Ecriture complète du programme

A partir du modèle général de programmation linéaire présenté en (2.9) et de la fonction objectif du (6.1), nous pouvons écrire l'ensemble du programme en (6.2).

$$\left\{ \begin{array}{l} z_{\min} = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) + \sum_{\theta=1}^T \sum_{r=1}^R \beta_{r\theta} (V_{r\max} - V_r(\theta)) \\ \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, V_r(\theta) = V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^{\theta} C_r(t) \\ \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, V_{r\min} \leq V_r(\theta) \leq V_{r\max} \\ \forall r \in \langle 1; R \rangle, V_r(\theta) = V_{r\max} \\ \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \\ \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \forall i \in \langle 1; I_g \rangle, \alpha_i^g(\theta) \geq 0 \end{array} \right. \quad (6.2)$$

6.2.1.3 Résolution

La forme de (6.2) étant celle d'un programme linéaire classique, la résolution se fait par une méthode type simplexe. Les essais réalisés montrent rapidement la limite de ce genre de méthode. La difficulté rencontrée est celle du choix des $\beta_{r\theta}$ par rapport aux $\pi_i^g(\theta)$ préalablement déterminés.

Les méthodes proposées par la suite évitent le problème du choix relatif des coûts des deux parties de la fonction objectif. Les deux méthodes proposées, par les graphes et par la programmation linéaire, reposeront sur le même principe :

1. résolution du modèle classique permettant d'exhiber une solution parmi un éventuel ensemble de solutions équivalentes,
2. résolution d'un problème d'optimisation exclusivement parmi l'ensemble des solutions équivalentes.

Ce type de méthode nous permet d'être sûr de travailler dans le cadre des solutions équivalentes puisque les deux parties de l'optimisation sont bien distinctes.

6.2.2 Programmation par les graphes

6.2.2.1 Structure des coûts

Comme cela avait été exposé au paragraphe 2.1, les fonctions coûts sont portées par les arcs du graphe. Pour expliquer comment se présentent les solutions équivalentes dans le cadre de la modélisation par les graphes nous allons représenter figure 6.1 la

modélisation du réseau AEP de la figure 2.2 en insistant sur les périodes où les tarifs électriques sont identiques.

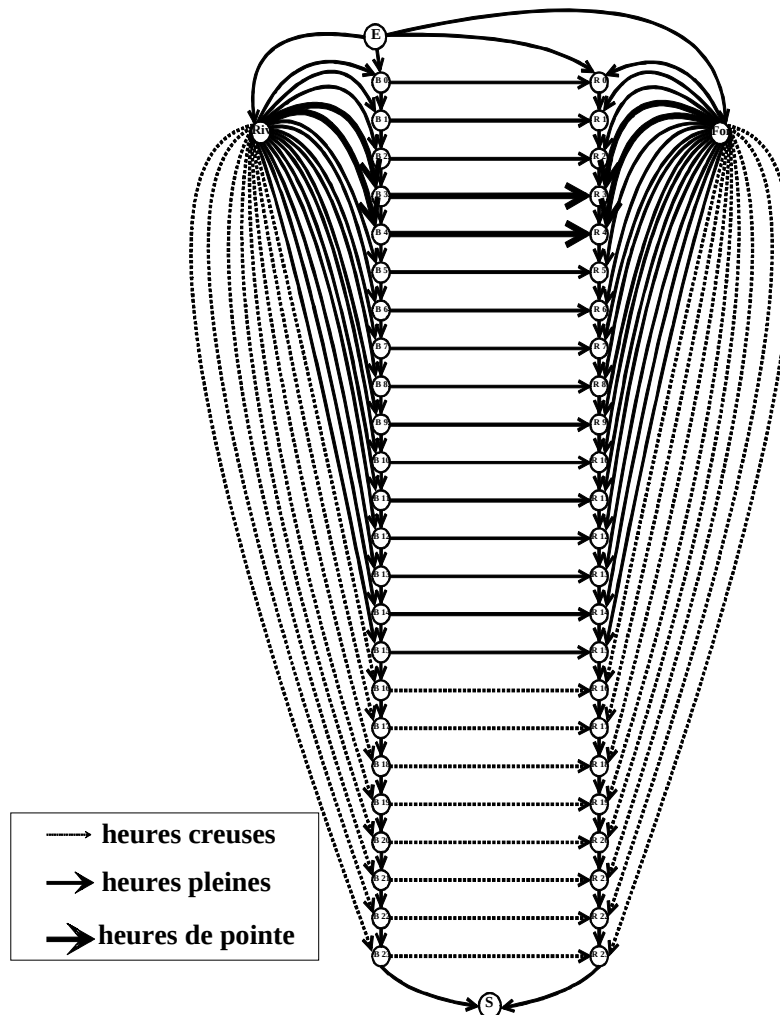


Figure 6.1 : Graphe avec mise en évidence des plages de tarifs électriques

Sur le graphe de la figure 6.1, les arcs représentant les consommations horaires n'ont pas été dessinés, étant entendu qu'ils seront saturés par l'optimisation. Sur ce graphe apparaissent des périodes de l'horizon d'optimisation où les tarifs électriques sont les mêmes. L'optimisation débutant à 6 heures, les tarifs électriques correspondent aux heures pleines en début d'horizon. Cette plage d'heures pleines est interrompue par une plage d'heures de pointe et suivie par les heures creuses de la nuit. Les coûts de production et de transport qui sont affectés aux arcs sont des fonctions affines des tarifs électriques. C'est pourquoi ces plages sont aussi des plages de coûts constants.

La recherche de flot maximal à coût minimal effectuée sur le graphe de la figure 6.1 donne une solution qui peut ne pas être unique. Ce qui apparaît en examinant le graphe est que toute solution équivalente aura la même répartition des flots par périodes de coûts identiques. Par contre à l'intérieur d'une même plage de coût, il est peut être équivalent de déplacer une partie du flot d'un arc vers un arc de la même famille.

6.2.2.2 Méthode proposée

Elle consiste à effectuer une première optimisation sur le graphe de la figure 6.1 puis à optimiser les trois sous-graphes représentés figure 6.2.

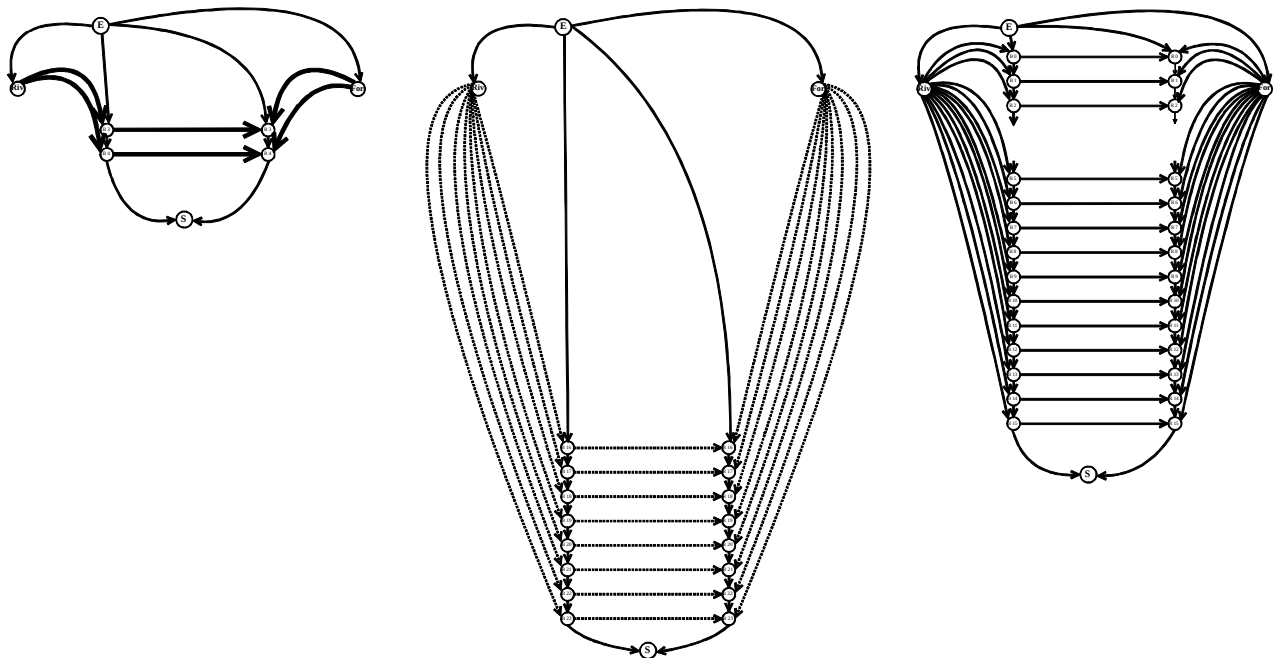


Figure 6.2 : Sous-graphes par plages de coût

6.2.2.2.1 Choix des capacités Pour chacun des sous-graphes il est nécessaire d'imposer les flots d'entrée et de sortie de façon à ce que la solution obtenue vérifie bien les mêmes propriétés que celle du graphe initial. Ceci se fera à l'aide des arcs issus du sommet d'entrée (noté E) de chacun des sous-graphes ainsi que les arcs aboutissant au sommet de sortie (noté S). Sur ces arcs nous modéliserons une capacité égale à la capacité minimale et valant le flot total de la plage de coût obtenu lors de l'optimisation du graphe de la figure 6.1.

6.2.2.2.2 Choix des coûts Lors de l'optimisation des sous-graphes, nous ne tenons plus compte des coûts de transit modélisés sur le graphe de la figure 6.1, étant entendu que la méthode proposée permet de travailler parmi les solutions optimales au sens de ce premier graphe. Il faut alors mettre en place d'autres coûts traduisant le deuxième objectif. Si nous voulons trouver un flot qui maximise les volumes horaires du réservoir de tête, nous donnerons un coût à chaque arc de transfert des stocks dans les réservoirs en choisissant des coûts plus faibles pour ce réservoir de tête.

Nous ne détaillons pas davantage cette méthode par les graphes car les choix des nouveaux coûts ainsi que les résultats obtenus sont comparables à ceux que nous présenterons au paragraphe 6.2.3. L'intérêt de la méthode par les graphes est, une fois de plus, la facilité de représentation visuelle des phénomènes de trajectoires équivalentes mises en jeu.

6.2.3 Programmation linéaire

6.2.3.1 Principe de la méthode

Il s'agit là aussi d'une optimisation en deux étapes.

La première étape consiste à optimiser le programme linéaire que nous avons déjà présenté dans l'équation (2.9). A l'issue de cette première optimisation nous obtenons la valeur des différentes commandes horaires par le biais des $\alpha_i^g(\theta)$ ainsi que la valeur de la fonction objectif minimisée.

La deuxième étape consiste alors à construire un programme linéaire proche de celui de la première étape. Il y a deux différences entre ces deux programmes : la valeur de la fonction objectif obtenue à l'issue de la première étape devient une contrainte dans la deuxième étape et une nouvelle fonction objectif est construite pour répondre au deuxième objectif. La solution trouvée à la deuxième étape fera bien partie des solutions équivalentes puisqu'elle respectera la nouvelle contrainte d'égalité.

Comme nous l'avons vu dans l'introduction du présent chapitre, nous pouvons dégager deux grandes catégories d'objectifs : la minimisation du nombre de mise en route des organes du réseau ou la maximisation des niveaux dans certains réservoirs.

6.2.3.2 Limitation du nombre de mises en route

Pour dénombrer les mises en route d'un organe, et pouvoir ainsi minimiser ce nombre, il est nécessaire de construire un modèle proche de celui que nous avons présenté au paragraphe 5.3.2.3.

Nous n'avons ici que deux tests à effectuer par organe sur lequel nous voulons compter les mises en route. Nous continuons à appeler Ω l'ensemble défini par $\Omega(i) = \left\{ i \in \langle 1; I \rangle / \left(\vec{K}_i \right)_{k_0} = 1 \right\}$. Les deux tests par organe concerné s'écrivent alors :

$$\begin{aligned} & - \sum_{i \in \Omega(i)} \alpha_i^g(\theta) > 0 \text{ ou } \sum_{i \in \Omega(i)} \alpha_i^g(\theta) = 0, \\ & - \sum_{i \in \Omega(i)} \alpha_i^g(\theta) < 1 \text{ ou } \sum_{i \in \Omega(i)} \alpha_i^g(\theta) = 1. \end{aligned}$$

Nous créons alors deux variables intermédiaires réelles $u(\theta)$ et $v(\theta)$ par ensemble Ω . Les tests s'écrivent alors :

$$\begin{aligned} & - \text{Si pour } \theta_0 \in \langle 1; T \rangle, \sum_{i \in \Omega(i)} \alpha_i^g(\theta) > 0 \text{ alors } u(\theta) \geq 2 \text{ sinon } u(\theta) \leq 1, \\ & - \text{Si pour } \theta \in \langle 2; T \rangle, \sum_{i \in \Omega(i)} \alpha_i^g(\theta - 1) < 1 \text{ alors } v(\theta) \geq 2 \text{ sinon } v(\theta) \leq 1. \end{aligned}$$

La transformation de ces implications (toujours à l'aide de la méthode décrite au paragraphe 5.3.2.1) nécessite l'introduction de 4 variables binaires par pas de temps. Par contre la suite ne demande par l'ajout d'autres variables binaires puisqu'il suffit maintenant de minimiser la somme $\sum_{\Omega} (u(\theta) + v(\theta))$. C'est cette expression qui constitue notre fonction objectif.

6.2.3.3 Optimisation des niveaux

6.2.3.3.1 Choix de l'objectif Il nous faut tout d'abord expliciter l'objectif d'un point de vue pratique. On peut, en fait, voir deux objectifs différents :

1. maximiser les volumes sur l'ensemble des réservoirs et pour tout pas de temps. Cet objectif revient à faire fonctionner les commandes le plus tôt possible à l'intérieur des plages de coût identique.
2. Maximiser les volumes en privilégiant certains réservoirs à certaines heures. En particulier nous pourrions choisir une fonction objectif dans laquelle figurent tous les volumes à toutes les heures mais où seront introduits des coefficients de pondération.

Voyons équation (6.3), l'écriture de la fonction objectif dans le deuxième cas.

$$z_{\min} = \sum_{\theta=1}^T \sum_{r=1}^R \beta_r(\theta) \cdot (V_{r \max} - V_r(\theta)) \quad (6.3)$$

Il est aisé de revenir au premier cas en prenant $\forall r \in \langle 1; R \rangle, \forall \theta \in \langle 1; T \rangle, \beta_r(\theta) = 1$, ce que nous ferons plus loin dans ce paragraphe.

Il est possible de développer l'équation (6.3) pour ne faire apparaître dans la fonction objectif, que les inconnues $\alpha_i^g(\theta)$. C'est ce que nous pouvons voir dans l'équation (6.4).

$$z_{\min} = \sum_{\theta=1}^T \sum_{r=1}^R \beta_r(\theta) \cdot \left(V_{r \max} - \left(V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^{\theta} C_r(t) \right) \right) \quad (6.4)$$

Le but étant la minimisation d'une fonction, nous pouvons nous passer des termes constants et écrire que l'équation (6.4) est équivalente à (6.5).

$$z_{\max} = \sum_{\theta=1}^T \sum_{r=1}^R \beta_r(\theta) \cdot \left(\sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) \right) \quad (6.5)$$

C'est bien l'expression de l'équation (6.5) qui constitue notre fonction objectif dans la deuxième étape de l'optimisation.

6.2.3.3.2 Choix des coefficients de pondération Dans l'expression (6.5), il nous reste à choisir la valeur des coefficients de pondération $\beta_r(\theta)$. Il est possible de privilégier certaines heures et certains réservoirs en fonction de l'expérience acquise par les gestionnaires du réseau sur lequel porte l'optimisation. On peut, par exemple, privilégier un réservoir desservant une zone sensible de par la présence de consommateurs plus fragiles de type hôpital et ce aux heures qui précèdent la pointe de fin de journée.

Une autre façon de voir les choses est de raisonner sur l'autonomie des réservoirs en fonction du pas de temps. Les coefficients de pondération sont alors définis par l'expression (6.6).

$$\forall r \in \langle 1; R \rangle, \forall \theta \in \langle 1; T \rangle, \beta_r(\theta) = \frac{C_r(\theta)}{V_{r \max}} \quad (6.6)$$

L'expression de l'équation (6.6) signifie que plus un réservoir aura une autonomie faible, plus le coefficient $\beta_r(\theta)$ sera élevé et plus ce volume à ce pas de temps aura du poids au sein de la fonction objectif.

6.2.3.3.3 Pondération identique Si l'on ne souhaite pas tenir compte de différences entre les réservoirs ou le temps ($\forall r \in \langle 1; R \rangle, \forall \theta \in \langle 1; T \rangle, \beta_r(\theta) = 1$), nous pouvons simplifier l'écriture (6.5) en utilisant une propriété des constantes $Q_{ri}^g(\theta)$. En effet, lorsque la commande $\vec{K}_i^g$ concerne un organe situé entre deux ou plusieurs réservoirs, nous avons la propriété de l'équation (6.7).

$$\sum_{r=1}^R Q_{ri}^g(\theta) = 0 \quad (6.7)$$

Si l'on note G' l'ensemble des groupes de commande ne couvrant que des zones de ressource, nous pouvons écrire la fonction objectif sous la forme (6.8).

$$z_{\max} = \sum_{\theta=1}^T \sum_{r=1}^R \sum_{t=1}^{\theta} \sum_{g \in G'} \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) \quad (6.8)$$

On peut finalement écrire l'équation (6.9).

$$z_{\max} = \sum_{j=i}^T \gamma_j \sum_{\theta=1}^T \sum_{r=1}^R \sum_{g \in G'} \sum_{i=1}^{I_g} \alpha_i^g(\theta) Q_{ri}^g(\theta) \quad (6.9)$$

Dans l'équation (6.9), les coefficients γ_j sont définis par $\forall j \in \langle 1; T \rangle, \gamma_j = T - j$

Ceci signifie que la fonction objectif retenue se traduit par la maximisation des $\alpha_i^g(\theta)$ correspondants à des ressources et ce en début d'horizon d'optimisation (du fait de la forme des constantes γ_j). Cette interprétation n'est pas surprenante puisque pour maximiser l'ensemble des volumes sur l'horizon d'optimisation, il est intéressant de produire l'eau le plus tôt possible.

6.2.3.4 Ecriture complète du modèle

6.2.3.4.1 Première étape Nous pouvons reprendre une forme déjà présentée de programme linéaire comme l'équation (2.9).

$$\left\{ \begin{array}{l} z_{\min}^1 = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) \\ \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^{\theta} C_r(t) \leq V_{r \max} \\ \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^{\theta} C_r(t) \geq V_{r \min} \\ \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^T C_r(t) = V_{r \max} \\ \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \\ \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \forall i \in \langle 1; I_g \rangle, \alpha_i^g(\theta) \geq 0 \end{array} \right. \quad (6.10)$$

Nous notons $\tilde{z}^1$ la valeur de la fonction objectif à l'optimum.

6.2.3.4.2 Deuxième étape Le programme linéaire de la deuxième étape se présente sous la forme (6.11). Nous avons choisi de présenter la forme générale de la fonction

objectif sans tenir compte des simplifications qu'apporte le fait de prendre les coefficients de pondération tous égaux à 1.

$$\left\{ \begin{array}{l} z_{\max}^2 = \sum_{\theta=1}^T \sum_{r=1}^R \beta_r(\theta) \cdot \left(\sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) \right) \\ \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^{\theta} C_r(t) \leq V_{r \max} \\ \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^{\theta} C_r(t) \geq V_{r \min} \\ \forall r \in \langle 1; R \rangle, V_r(0) + \sum_{t=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(t) - \sum_{t=1}^T C_r(t) = V_{r \max} \\ \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) = \tilde{z}^1 \\ \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \\ \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \forall i \in \langle 1; I_g \rangle, \alpha_i^g(\theta) \geq 0 \end{array} \right. \quad (6.11)$$

6.3 Conclusion

Le constat initial est la particularité numérique de notre problème qui implique souvent l'existence de solutions de même coût sur une journée. Il est donc possible de hiérarchiser ces solutions que l'on peut caractériser d'équivalentes. La difficulté de maîtriser les poids respectifs des deux termes d'une programmation multi-objectif nous amène à proposer des méthodes en deux étapes. La présentation de cette méthode par la programmation par les graphes permet de bien visualiser la notion de trajectoires équivalentes. La programmation linéaire permet de construire des modèles et des algorithmes plus faciles à mettre en oeuvre.

Chapitre 7

Influence de l'état du réseau sur les coûts et les capacités de transit

7.1 Ecriture du problème

Quel que soit le modèle d'optimisation utilisé jusqu'alors, nous avons toujours considéré que les coûts de production et de transport de l'eau étaient connus avant le calcul d'optimisation. De la même façon, nous considérons que les débits transitant entre ouvrages étaient connus en ne dépendant éventuellement que des consommations en route.

Le phénomène de dépendance vis-à-vis de l'état du réseau existe en particulier lorsque l'entrée dans un réservoir se fait par le fond de celui-ci. La charge à vaincre ne sera pas la même si le réservoir est plein ou vide. Cela se traduira par un débit entrant plus fort si le réservoir est vide. De même la puissance consommée par une pompe sera plus importante si le réservoir est plein, ce qui se traduira par un coût de transit plus important.

D'un point de vue modèle d'optimisation, le problème rencontré revient à définir des paramètres du modèle dépendant du résultat du calcul. Dans un premier temps, seule la programmation dynamique semble pouvoir répondre à ce type de difficultés. Nous avons alors voulu nous affranchir de cette méthode qui entraînerait des temps de calcul importants en proposant deux autres méthodes.

7.2 Formulation par programmation linéaire en nombres entiers

Nous cherchons à traduire les niveaux dans les réservoirs par des variables entières et à les intégrer directement au modèle linéaire du type de celui présenté en équation (2.9). Nous ne présentons ici que l'influence des volumes par rapport aux débits entrant dans les réservoirs.

7.2.1 Modification des débits entrant

Au paragraphe (2.2.3) nous avons défini les débits entrant dans chaque réservoir par l'équation (2.7) reprise ici en équation (7.1).

$$\forall r \in \langle 1; R \rangle, Q_r(\theta) = \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) Q_{ir}^g(\theta) \quad (7.1)$$

Pour tenir compte de l'influence des niveaux dans les réservoirs sur le débit entrant, nous allons maintenant écrire la relation de l'équation (7.2).

$$Q_{ir}^g(\theta) = f(V_j(\theta)) \quad (7.2)$$

Dans cette équation, les variables $V_j(\theta)$ représentent les volumes des différents réservoirs j aux pas de temps θ .

7.2.2 Illustration de la dépendance

Voyons sur le réseau exemple de la figure 7.1, l'illustration du phénomène d'influence des niveaux sur les transits.

Sur la figure 7.1 les cotes des réservoirs R1 et R2 sont choisies de façon à faire apparaître une influence des niveaux sur les transits entre ouvrages. Voyons, figure 7.2, le débit entrant dans R2 en fonction de la différence de cote entre RTETE et R2.

Par une régression on obtient l'équation :

$$Q_{RTETE \rightarrow R2} = -0,26 * (V_{RTETE} - V_{R2})^2 + 15,5 * (V_{RTETE} - V_{R2}) + 56,91 \quad (7.3)$$

Nous approximations cette écriture par une régression linéaire de plusieurs variables qui s'écrit comme l'équation (7.4).

$$Q_{RTETE \rightarrow R2} = 10,53 * V_{RTETE} - 10,53 * V_{R2} + 78,77 \quad (7.4)$$

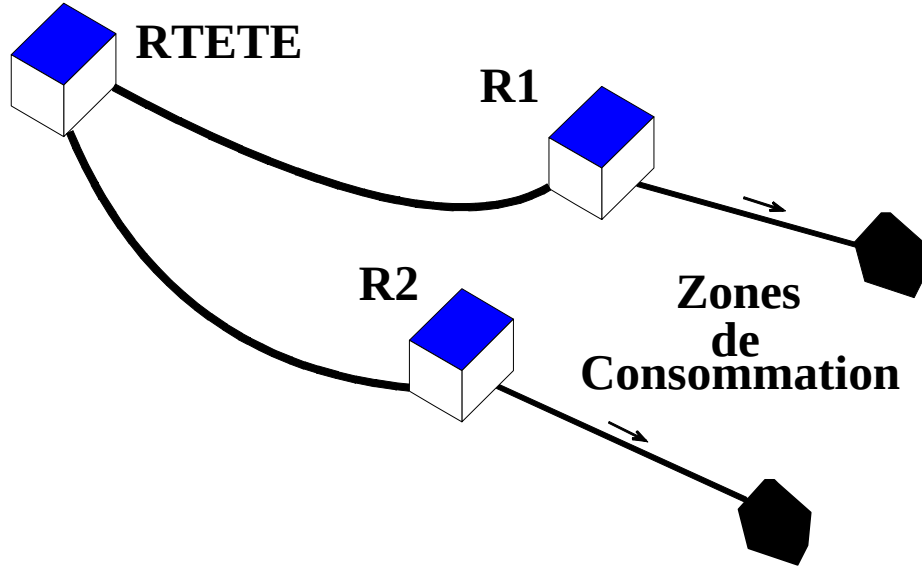


Figure 7.1 : Réseau AEP illustrant l'influence des niveaux sur les transits

7.2.3 Méthode proposée

7.2.3.1 Discrétisation des volumes

Pour intégrer le phénomène mis en évidence dans un programme linéaire, nous procédons à une discrétisation des volumes dans chaque réservoir en définissant pour chaque $r \in \langle 1; R \rangle$ $D+2$ constantes S_r^d avec en particulier $S_r^0 = V_{r \min}$ et $S_r^{D+1} = V_{r \max}$ avec $V_{r \min}$ le volume correspondant au niveau radier du réservoir r et $V_{r \max}$ le volume correspondant au trop plein de ce même réservoir.

L'équation (7.2) se transforme alors de la façon suivante :

$$Q_{ir}^g(\theta) = a_{ir}^g + \sum_{j=1}^R b_{irj}^g(\theta) \quad (7.5)$$

$$b_{irj}^g(\theta) = \sum_{d=1}^D b_{irjd}^g(\theta) \quad (7.6)$$

Les différents coefficients $b_{irjd}^g(\theta)$ s'expriment par les relations suivantes de l'équation (7.7).

$$b_{irj1}^g(\theta) = c_{irj1}^g \text{ et } \forall k \in \langle 1; D-1 \rangle \begin{cases} \text{Si } V_j(\theta) < S_j^k \text{ alors } b_{irj(k+1)}^g(\theta) = 0 \\ \text{Si } V_j(\theta) > S_j^k \text{ alors } b_{irj(k+1)}^g(\theta) = c_{irj(k+1)}^g \end{cases} \quad (7.7)$$

Les coefficients c étant des constantes obtenues à partir d'une régression multiple

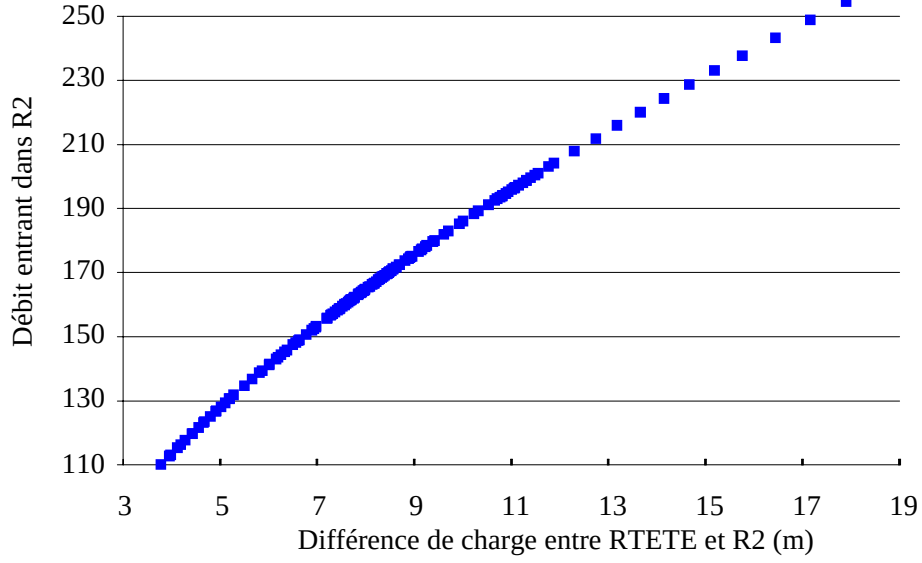


Figure 7.2 : Influence des niveaux sur les débits

en transformant une fonction continue du type (7.4) en une fonction constante par morceaux.

La transformation des implications de l'équation (7.7) se fait par la méthode présentée au paragraphe 5.3.2.1 en introduisant des variables booléennes $y_{irjk}^g(\theta)$ et $v_{irjk}^g(\theta)$ ainsi que la constante M comme développé à l'équation (7.8).

$$\forall k \in \langle 1; D-1 \rangle \left\{ \begin{array}{l} S_j^k - V_j(\theta) \leq M \times (1 - y_{irjk}^g(\theta)) \\ b_{irj(k+1)}^g(\theta) \leq M \times y_{irjk}^g(\theta) \\ -b_{irj(k+1)}^g(\theta) \leq M \times y_{irjk}^g(\theta) \\ V_j(\theta) - S_j^k \leq M \times (1 - v_{irjk}^g(\theta)) \\ b_{irj(k+1)}^g(\theta) - c_{irj(k+1)}^g(\theta) \leq M \times v_{irjk}^g(\theta) \\ c_{irj(k+1)}^g(\theta) - b_{irj(k+1)}^g(\theta) \leq M \times v_{irjk}^g(\theta) \end{array} \right. \quad (7.8)$$

7.2.3.2 Ecriture complète du programme

Le programme complet d'optimisation s'écrit alors sous la forme de l'équation (7.9).

$$\left\{ \begin{array}{l} z_{\min} = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) \text{ sous les contraintes :} \\ \forall \theta \in \langle 1; T \rangle, \forall r \in \langle 1; R \rangle, V_r(\theta) = V_r(\theta - 1) + \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) Q_{ri}^g(\theta) - \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) C_{ri}^g(\theta) \\ \forall \theta \in \langle 1; T \rangle, \forall r \in \langle 1; R \rangle, \forall g \in \langle 1; G \rangle, \forall i \in \langle 1; I_g \rangle, Q_{ir}^g(\theta) = a_{ri}^g + \sum_{j=1}^R \left(c_{rij1}^g + \sum_{d=2}^D b_{rijd}^g(\theta) \right) \\ \forall \theta \in \langle 1; T \rangle, \forall r \in \langle 1; R \rangle, \forall g \in \langle 1; G \rangle, \forall i \in \langle 1; I_g \rangle, \forall k \in \langle 1; D - 1 \rangle \left\{ \begin{array}{l} S_j^k - V_j(\theta) \leq M \times (1 - y_{irjk}^g(\theta)) \\ b_{irj(k+1)}^g(\theta) \leq M \times y_{irjk}^g(\theta) \\ -b_{irj(k+1)}^g(\theta) \leq M \times y_{irjk}^g(\theta) \\ V_j(\theta) - S_j^k \leq M \times (1 - v_{irjk}^g(\theta)) \\ b_{irj(k+1)}^g(\theta) - c_{irj(k+1)}^g(\theta) \leq M \times v_{irjk}^g(\theta) \\ c_{irj(k+1)}^g(\theta) - b_{irj(k+1)}^g(\theta) \leq M \times v_{irjk}^g(\theta) \end{array} \right. \\ \forall \theta \in \langle 1; T - 1 \rangle, \forall r \in \langle 1; R \rangle, V_{r\min} \leq V_r(\theta) \leq V_{r\max} \\ \forall r \in \langle 1; R \rangle, V_r(T) = V_{r\max} \\ \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \end{array} \right. \quad (7.9)$$

Le programme (7.9) n'est malheureusement pas un programme linéaire en nombres mixtes. La fonction objectif est bien linéaire puisque nous avons choisi de ne pas tenir compte de l'influence des niveaux dans les réservoirs sur les coûts unitaires $\pi_i^g(\theta)$. La première contrainte n'est, par contre, pas linéaire. Les termes du type $\alpha_i^g(\theta) Q_{ri}^g(\theta)$ constituent, en effet, la multiplication de deux inconnues entre elles. Le programme présenté est donc un programme en variables mixtes d'une fonction objectif linéaire et de contraintes en partie quadratique. La résolution d'un tel problème par cette méthode s'avère impossible.

7.3 Résolution par la programmation linéaire successive

7.3.1 Principe de la méthode

Dans le paragraphe 1.2.2.4.2 nous avons détaillé plusieurs méthodes de programmation linéaire successive.

7.3.2 Ecriture non-linéaire du problème d'optimisation

Sans utiliser les simplifications proposées au paragraphe 7.2, le programme d'optimisation permettant de tenir compte de l'influence des niveaux dans les réservoirs sur les coûts et les transits se présente sous la forme de l'équation (7.10).

$$\left\{ \begin{array}{l} z_{\min} = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(V_1(\theta-1), \dots, V_R(\theta-1)) \text{ sous les contraintes :} \\ \forall \theta \in \langle 1; T \rangle, \forall r \in \langle 1; R \rangle, V_r(\theta) = \begin{array}{l} V_r(\theta-1) \\ + \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) Q_{ri}^g(V_1(\theta-1), \dots, V_R(\theta-1)) \\ - \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) C_{ri}^g(\theta) \end{array} \\ \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, V_{r\min} \leq V_r(\theta) \leq V_{r\max} \\ \forall r \in \langle 1; R \rangle, V_r(T) = V_{r\max} \\ \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \end{array} \right. \quad (7.10)$$

Dans cette écriture (7.10), les fonctions π_i^g et Q_{ri}^g permettent de traduire l'influence des différents niveaux. Elles peuvent prendre des formes linéaires ou non linéaires.

7.3.3 Convergence générale du problème

Les références citées au paragraphe 1.2.2.4.2 aboutissent toutes à l'impossibilité de prouver la convergence générale de la méthode de programmation linéaire successive. Dans (Minoux, 1983a), l'auteur donne un exemple intéressant de non-convergence. Il se place dans le cadre d'un programme à fonction objectif non-linéaire et à contraintes linéaires et examine l'une des deux grandes méthodes envisagées, l'approximation tangentielle qui consiste à approximer la fonction objectif par une fonction linéaire. Le contre exemple proposé peut être représenté sur la figure 7.3.

Ce que l'on observe sur le dessin de la figure 7.3 est un phénomène d'oscillation entre deux solutions. Si l'on débute la recherche de la solution en un point quelconque du domaine des admissibles x^0 , la première solution trouvée après la première linéarisation est x^1 , la deuxième est x^2 puis l'on retrouve x^1 puis indéfiniment ces deux solutions en alternance.

Le problème général de non-convergence est alors abordé par des techniques algorithmiques. Minoux les classe en deux grandes catégories :

- la limitation des déplacements,

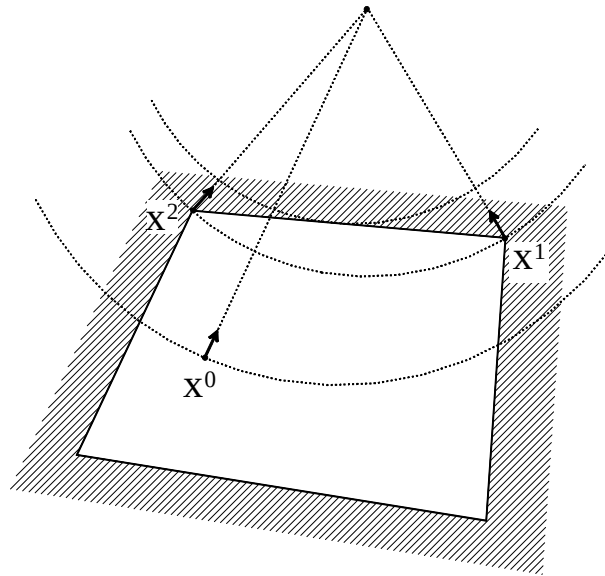


Figure 7.3 : Exemple de linérisation successive présenté dans (Minoux, 1983a)

- l'utilisation de points multiples de linéarisation.

Minoux signale deux types de problèmes où la convergence des méthodes de linéarisation peut être démontrée. Le premier type est un problème où toutes les contraintes sont linéaires. Dans ce cas, on peut appliquer la méthode de Frank et Wolf convergente pour toute fonction objectif continûment différentiable. Le principe de la méthode de Frank et Wolf (1956) consiste à utiliser le gradient de la fonction objectif au point courant. Le second type de problème où les méthodes itératives convergent est celui des problèmes convexes à fonction objectif et contraintes non linéaires. La méthode alors appliquée est celle des plans sécants de Kelley (1960). Cette-ci consiste à rajouter une contrainte linéaire à chaque itération de façon à réduire le domaine des admissibles.

Comme au paragraphe 7.2 nous ne prenons en compte que l'influence des niveaux des réservoirs sur les débits transitant sur le réseau et pas sur les coûts de ces transits. Le problème général que nous avons présenté en (7.10) est alors un programme à fonction objectif linéaire et à contraintes non linéaires. Pour savoir dans quel cas nous nous situerons dans le cadre d'un programme convexe il convient donc d'examiner la convexité du domaine des admissibles. En particulier, nous savons que si nous écrivons le programme sous la forme de l'équation (7.11), nous pourrions écrire une condition

suffisante de convexité.

$$\left\{ \begin{array}{l} \text{Min } f(x) \\ \text{sous les contraintes} \\ g_i(x) \leq 0, i \in \langle 1, m \rangle \end{array} \right. \quad (7.11)$$

Si toutes les fonctions $g_i(x)$ sont convexes le programme de l'équation (7.11) sera bien convexe. Mettons le problème (7.10) sous la forme (7.12).

$$\left\{ \begin{array}{l} z_{\min} = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(\theta) \text{ sous les contraintes :} \\ (1) \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, \\ V_r(0) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(V(t-1)) - \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) C_{ri}^g(t) - V_{r\max} \leq 0 \\ (2) \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, \\ -V_r(0) - \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(V(t-1)) + \sum_{t=1}^{\theta} \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) C_{ri}^g(t) + V_{r\min} \leq 0 \\ (3) \forall r \in \langle 1; R \rangle, \\ -V_r(0) - \sum_{t=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) Q_{ri}^g(V(t-1)) + \sum_{t=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(t) C_{ri}^g(t) + V_{r\max} \leq 0 \\ (4) \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \end{array} \right. \quad (7.12)$$

Dans ce programme les contraintes de type (1), (2) et (3) sont non-linéaires. Dans le cas où il est possible d'écrire les fonctions Q_{ri}^g comme des combinaisons linéaires des volumes des différents réservoirs (nous avons vu l'exemple des équations (7.3) et (7.4) les différentes fonctions g_i seront des fonctions polynomiales du second ordre. Ceci signifie que les fonctions de type g_i du programme (7.10) seront convexes si et seulement si tous les coefficients de degré 2 des fonctions g_i sont positifs ou nuls. Or nous voyons dans le programme (7.12) que les mêmes fonctions $Q_{ri}^g(V(t-1))$ interviennent dans les contraintes de type (1) avec un signe "+" et dans les contraintes de type (2) avec un signe "-".

La condition suffisante d'obtention d'un espace des admissibles convexe n'est donc pas réalisée. En fait cette condition est mise en défaut de deux façons différentes :

1. par la présence systématique de deux fonctions de signes opposées dans les contraintes de type (1) et (2) comme nous venons de le voir,
2. par la présence de fonctions g_i qui sont non convexes lorsque les fonctions $Q_{ri}^g(V(t-1))$ sont des polynômes de degré un de plusieurs variables avec des signes différents pour chacune des variables (comme la fonction définie dans (7.4) par exemple).

La condition de convexité que nous venons d'étudier est bien une condition suffisante. Il reste donc possible que le problème soit néanmoins convexe mais il semble difficile de caractériser les réseaux pour lesquels le programme est convexe. La difficulté rencontrée ici de mettre en évidence une typologie des réseaux pour lesquels une méthode de type "plans sécants de Kelley" converge, nous amène à proposer une méthode plus simple et à étudier sur des exemples la limite de la convergence de celle-ci.

7.3.4 Linéarisation au premier ordre

En nous inspirant de différentes références ((Minoux, 1983a), (Palacios-Gomez, 1982), etc), nous proposons une méthode de programmation linéaire successive où la linéarisation se fait au premier ordre. Nous linéarisons le programme présenté dans la relation (7.10) en résolvant de façon successive le programme (7.13).

$$PL(V^i) \left\{ \begin{array}{l} z_{\min} = \sum_{\theta=1}^T \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) \pi_i^g(f_1(V_1^i(\theta-1), \dots, V_R^i(\theta-1))) \\ \text{sous les contraintes :} \\ \forall \theta \in \langle 1; T \rangle, \forall r \in \langle 1; R \rangle, \\ V_r^i(\theta) = V_r^i(\theta-1) \\ \quad + \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) Q_{ri}^g(f_2(V_1^i(\theta-1), \dots, V_R^i(\theta-1))) \\ \quad - \sum_{g=1}^G \sum_{i=1}^{I_g} \alpha_i^g(\theta) C_{ri}^g(\theta) \\ \forall \theta \in \langle 1; T-1 \rangle, \forall r \in \langle 1; R \rangle, V_{r\min} \leq V_r^i(\theta) \leq V_{r\max} \\ \forall r \in \langle 1; R \rangle, V_r^i(T) = V_{r\max} \\ \forall \theta \in \langle 1; T \rangle, \forall g \in \langle 1; G \rangle, \sum_{i=1}^{I_g} \alpha_i^g(\theta) = 1 \end{array} \right. \quad (7.13)$$

7.3.4.1 Principe de l'algorithme

Le principe de l'algorithme de programmation linéaire successive peut être représenté figure 7.4.

Le test de convergence de cette méthode doit permettre de vérifier que les résultats obtenus ne diffèrent que très peu d'une itération à la suivante. Nous utilisons le test de l'équation (7.14).

$$\sum_{\theta=1}^T \sum_{r=1}^R |V_r^{i+1}(\theta) - V_r^i(\theta)| \quad (7.14)$$

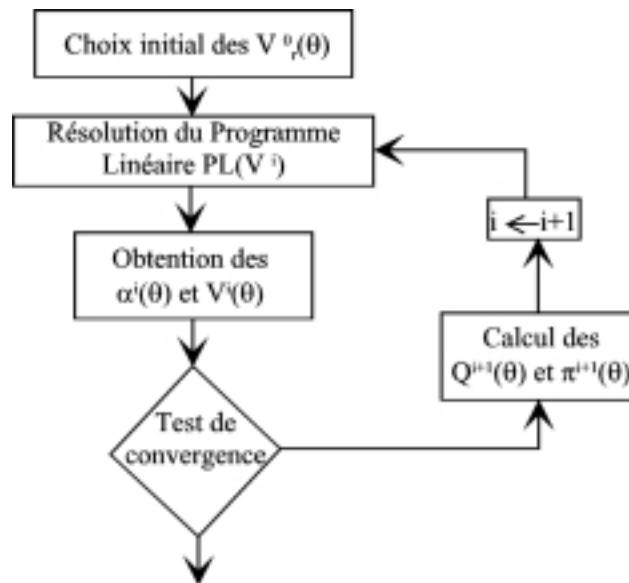


Figure 7.4 : Structure de l'algorithme de Programmation Linéaire Successive proposé

7.3.4.2 Mise en place sur un réseau exemple

7.3.4.2.1 Description de l'exemple Nous travaillons sur une partie du réseau d'eau potable du Syndicat Intercommunal des Eaux de la Région de Sarralbe. La zone qui nous intéresse peut être représentée sur la figure 7.5. Il s'agit de la partie ouest de l'ensemble du réseau. Les autres sous-réseaux ont été simplifiés et figurent comme des arrivées. Ce sous-réseau "Ouest" est donc alimenté par trois ressources : les arrivées Nord et Sud et la production de l'usine. Ces trois ressources sont à coûts différents. Les coûts dépendent uniquement de l'heure de la journée. La dépendance des coûts vis-à-vis de l'état des réservoirs n'est pas envisagée.

La suppression à la sortie de R2 se traduit par la mise en route d'une ou de deux pompes. Les réservoirs R2 et R5 permettent une séparation en groupes indépendants. Il y a alors 4 groupes comportant respectivement :

1. la vanne nord se traduisant par deux vecteurs de commande, vanne nord fermée et vanne nord ouverte,
2. la vanne sud se traduisant par deux vecteurs de commande, vanne sud fermée et vanne sud ouverte,
3. l'usine se traduisant par deux vecteurs de commande, usine fonctionnant et usine à l'arrêt,
4. toutes les autres commandes se traduisant par 15 vecteurs de commande que l'on

peut présenter dans le tableau 7.1.

Vecteur	Pompe 1	Pompe 2	Vanne R3	Vanne R4	Vanne R5
1	0	0	0	0	0
2	1	0	1	0	0
3	1	0	0	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	0	1	0	1
7	1	0	0	1	1
8	1	0	1	1	1
9	0	1	1	0	0
10	0	1	0	1	0
11	0	1	0	0	1
12	0	1	1	1	0
13	0	1	1	0	1
14	0	1	0	1	1
15	0	1	1	1	1

Tableau 7.1 : Vecteurs de commande du sous-réseau ouest

La valeur 1 figurant dans le tableau 7.1 traduit le fonctionnement des pompes ou l'ouverture des vannes.

En final le problème d'optimisation comportera donc 21 inconnues par pas de temps, soit 504 inconnues réelles $\alpha_i^g(\theta)$.

Les différents transits entre ouvrages dépendent de l'heure de la journée, de la consommation en route et de l'état des réservoirs. On peut donner en (7.15) l'exemple de l'écriture du débit entrant dans R4 lorsqu'une seule pompe est en marche et que seule la vanne R4 est ouverte.

$$Q_{entréeR4}(\theta) = 40,58 + 0,43 \times Conso(\theta) - 0,026 \times V_{R4}(\theta) \text{ avec } 0 \leq V_{R4}(\theta) \leq 380 \quad (7.15)$$

Tous les débits transitant entre les ouvrages sont exprimés par une relation du type de l'équation (7.15). On peut alors exécuter l'algorithme basé sur le principe de la figure 7.4.

7.3.4.2.2 Résultats Les résultats obtenus peuvent être représentés, figure 7.6, par l'évolution du critère de convergence de l'équation (7.14).

Le critère de convergence représenté commence par décroître rapidement puis, à partir de l'itération $n^{\circ}3$, celui-ci n'évolue plus de façon monotone. La convergence de la méthode est donc mise en défaut.

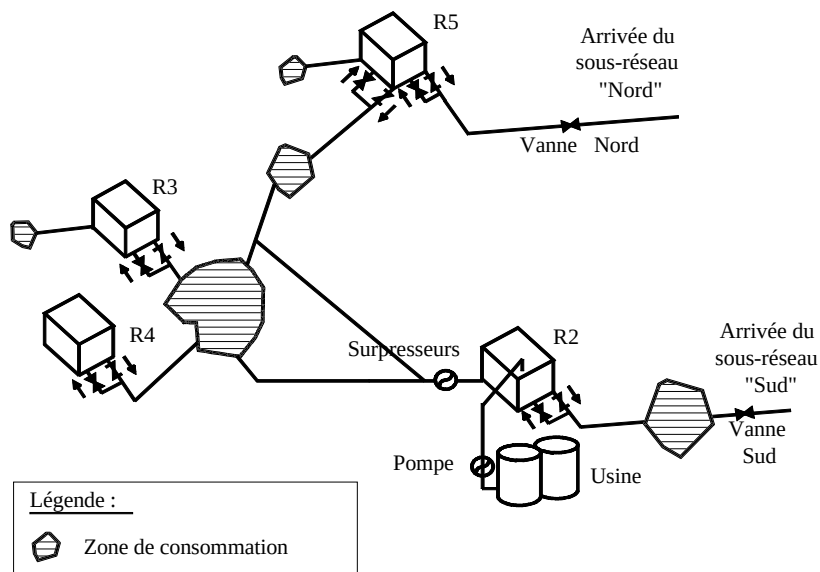


Figure 7.5 : Sous-réseau Ouest du SIER de Sarralbe

7.3.5 Utilisation des solutions équivalentes

7.3.5.1 Causes de la non-convergence

En examinant de plus près les raisons de la non-convergence de la méthode du paragraphe précédent, on peut dégager un phénomène important. Lors de la résolution de l'étape d'optimisation $PL(V^i)$, apparaissent des solutions équivalentes telles que nous avons pu les décrire au paragraphe 6.2.3.

En particulier, nous pouvons observer, figure 7.7, l'évolution de la trajectoire du niveau dans R5 au cours des 50 premières itérations.

Sur le graphique de la figure 7.7, nous pouvons voir qu'après trois itérations la trajectoire entre 22 heures et 6 heures est constante alors qu'elle varie entre 6 et 22 heures.

7.3.5.2 Méthode proposée

Nous reprenons le principe de l'algorithme présenté figure 7.4 en complétant le programme d'optimisation $PL(V^i)$ par un second programme linéaire permettant de retenir une seule solution parmi les solutions équivalentes. Le critère retenu pour la deuxième optimisation pourra être choisi parmi ceux évoqués au paragraphe 6.2.3.3 comme le volume dans un réservoir donné. Cette méthode devrait permettre de choisir, à chaque itération, la trajectoire favorisant ce deuxième critère.

Nous pouvons ainsi espérer que la fluctuation des trajectoires d'une itération à la

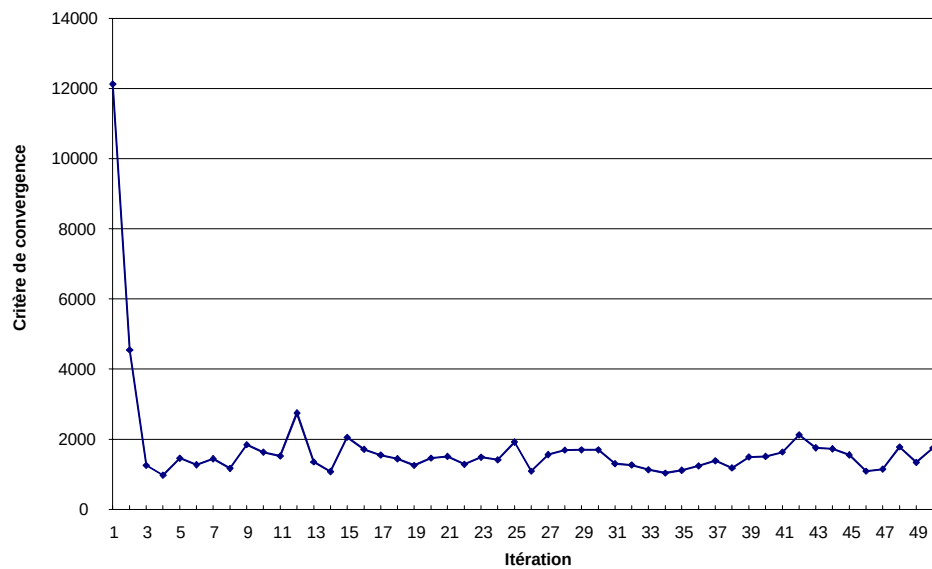


Figure 7.6 : Evolution du critère de convergence pour la linéarisation au premier ordre

suivante disparaisse et que le critère général de cette méthode de programmation linéaire successive tende vers zéro.

7.3.5.3 Résultats

La convergence de la méthode est cette fois obtenue. Le critère de convergence évolue comme détaillé dans le tableau 7.2.

Itération	1	2	3	4
Critère de convergence	16052,6	1115,1	20,8	0,2

Tableau 7.2 : Evolution du critère de convergence de la deuxième méthode

7.3.6 Limite de la convergence

7.3.6.1 Explication de la convergence sur le réseau exemple

Au paragraphe 7.3.5 nous avons construit une méthode aboutissant à la convergence. Néanmoins nous gardons à l'esprit le contre-exemple proposé par Minoux et représenté figure 7.3. De plus, nous avons montré au paragraphe 7.3.3 que le problème était en général non-convexe. Nous pouvons donc supposer que la convergence constatée est liée à la faible non-linéarité ou à la faible non-convexité de notre problème.

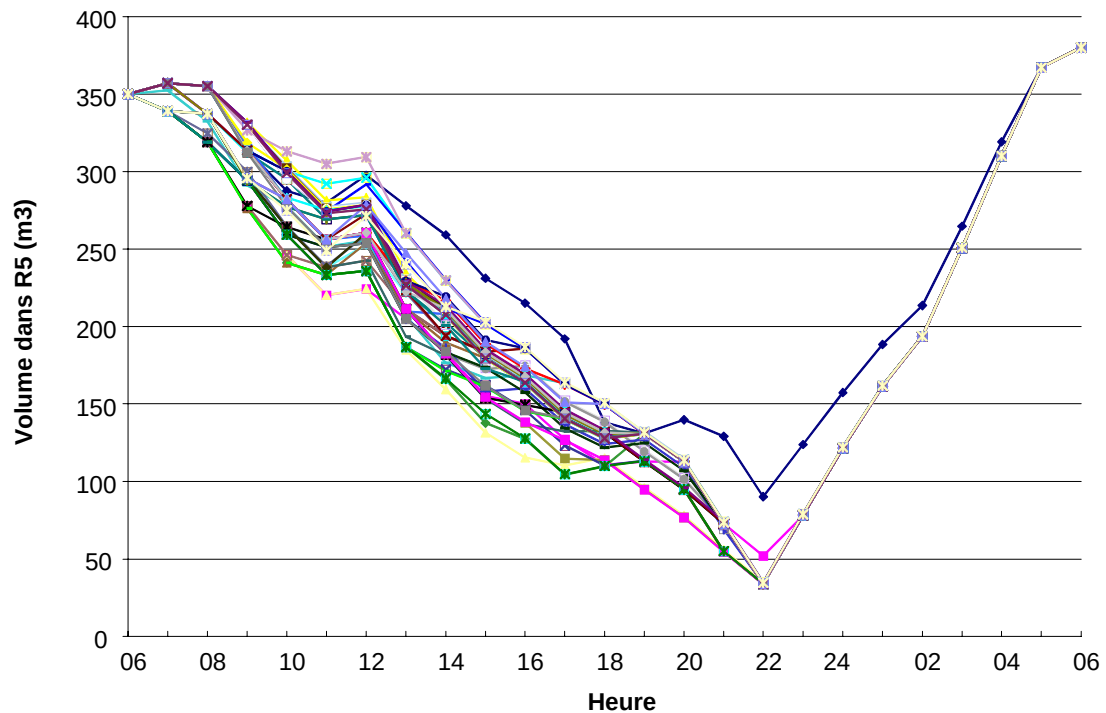


Figure 7.7 : Volume dans R5 au cours des itérations

7.3.6.2 Mise en évidence de la non-convergence

Pour vérifier l'explication proposée, nous transformons artificiellement l'exemple de la partie Ouest du réseau de Sarralbe en augmentant l'influence des hauteurs d'eau sur les débits transitant.

Concrètement, nous multiplions l'influence des hauteurs d'eau dans les réservoirs par un facteur de 10. Dans l'équation (7.15), par exemple, le coefficient 0,026 deviendra 0,26. Ceci permet de trouver des situations de réservoirs d'équilibre dans lesquels le débit peut changer de sens en fonction du volume contenu dans les réservoirs aval et amont.

Le critère de convergence défini à l'équation (7.14) décroît rapidement au cours de la méthode de programmation linéaire successive pour finalement se stabiliser à une valeur d'environ 18 m^3 à partir de la onzième itération. Si nous observons les volumes obtenus pour les quatre réservoirs nous voyons que pour deux d'entre eux la trajectoire est stable à partir de cette onzième itération alors que pour les deux autres la solution passe alternativement par deux trajectoires différentes. Ces deux trajectoires ne diffèrent qu'à 00 :00 heure comme nous pouvons le voir sur les graphiques 7.8 et 7.9. Sur ces graphiques nous avons représenté en trait plein les volumes obtenus à l'issue de chaque itération pair

et en trait pointillé les volumes à l'issue des itérations impaires.

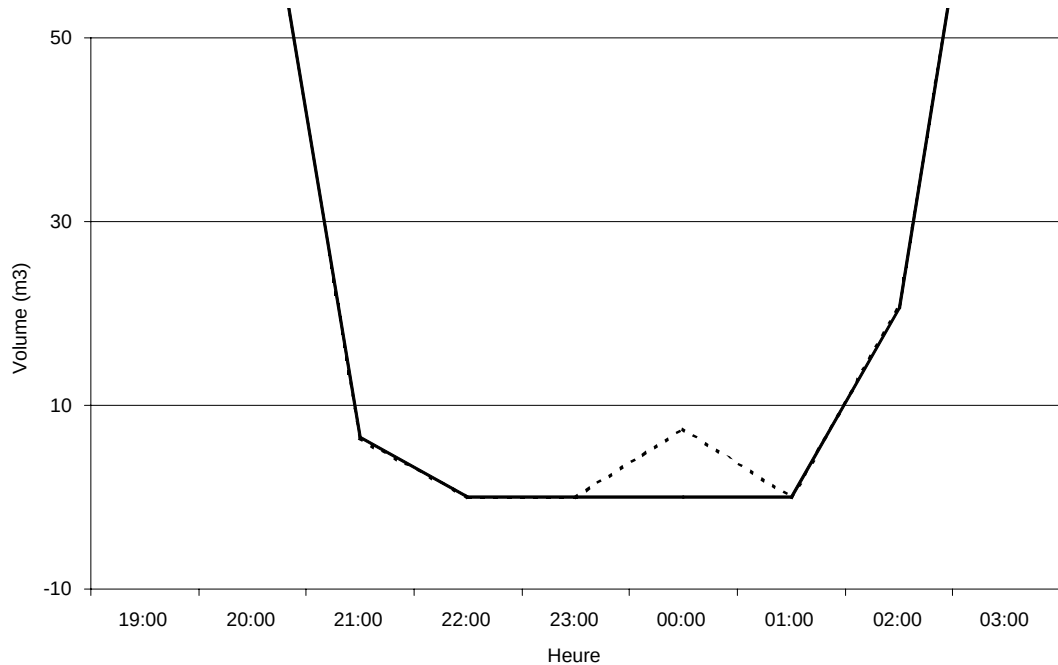


Figure 7.8 : Volume dans R2 entre 19 h et 3 h

Au-delà du simple constat de la proximité entre les deux trajectoires sur laquelle nous reviendrons, il est intéressant de constater que lorsqu'à l'issue d'une itération l'optimisation sature une contrainte de niveau minimal dans R2, elle relâche dans le même temps une contrainte de niveau minimal dans R5. C'est bien entendu le contraire qui se passe à l'itération suivante. Cela signifie que la non-convergence de ce contre-exemple est due à un saut d'une itération à la suivante entre deux solutions ne saturant pas les mêmes contraintes. Ceci semble dû à la non-convexité de notre domaine qui fait que chacune des sous-solutions est celle d'un sous-domaine localement convexe. Pour vérifier que nous ne pouvions quitter cette double solution pour trouver la vraie solution du problème de départ (qui n'est pas forcément l'une de ces solutions) nous avons transformé l'algorithme de façon à introduire des points intermédiaires entre ces deux solutions en nous inspirant des méthodes développées dans (Palacios-Gomez, 1982). Nous avons alors constaté que nous pouvions effectivement aboutir à des points intermédiaires mais qui ne constituent pas des solutions stables du problème ; celui-ci convergeant toujours vers les deux mêmes solutions.

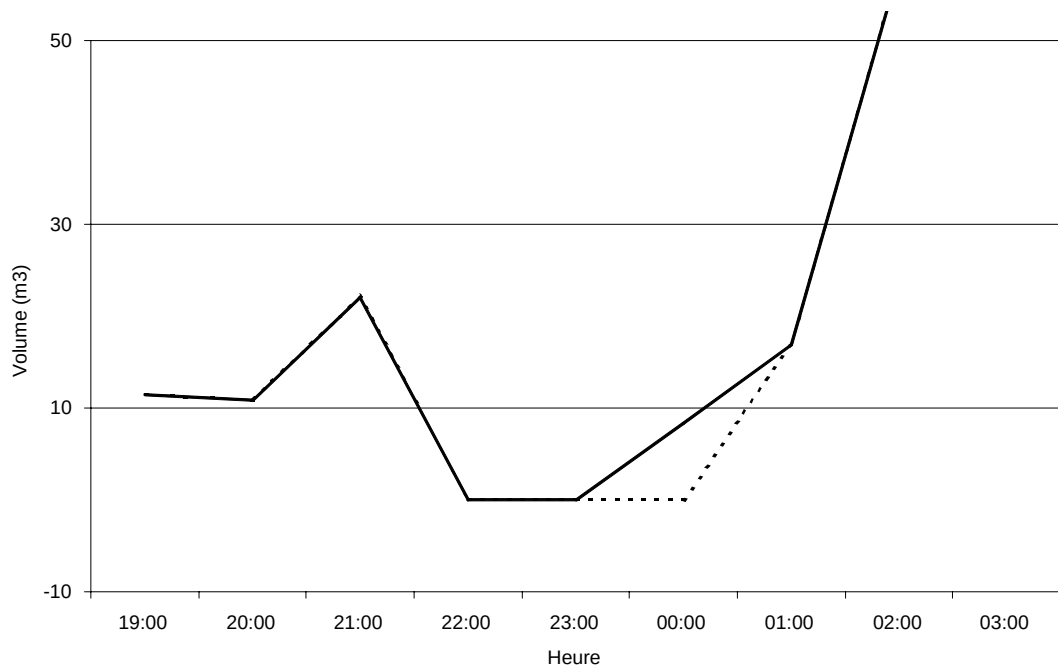


Figure 7.9 : Volume dans R5 entre 19 h et 3 h

7.3.6.3 Configuration hydraulique permettant la convergence

Nous avons déjà vu que la convergence sur un exemple réel s'expliquait par la faible non-linéarité (et donc non-convexité) des influences vis-à-vis de l'état du réseau. L'augmentation artificielle de cette influence amène à une non-convergence qui illustre la non-convexité du domaine des admissibles. Or nous avons vu au paragraphe 7.3.3 qu'il existait deux causes possibles de non-réalisation d'une condition suffisante d'optimalité. La première, l'existence de signes opposés dans les contraintes de niveaux min et max, est due à la structure générale de notre problème, elle est donc inévitable. De plus, elle n'est sans doute pas la cause de la non-convergence mise en évidence précédemment car elle crée plutôt une non-convexité générale qui ne concerne que peu la convergence finale de la méthode. Il est alors possible que la non-convexité soit plutôt due à l'existence de fonctions non-convexes dans les contraintes. Nous avons vu que cela pouvait être dû à l'existence d'influences multiples et de signes contraires de plusieurs réservoirs sur un seul transit entre ouvrages.

Dans l'exemple de la partie Ouest du réseau de Sarralbe, nous avons plusieurs transits faisant à la fois intervenir les niveaux dans R2 et R4. C'est pourquoi nous avons à nouveau modifié notre matrice de contraintes en supprimant toutes les influences du

7. Influence de l'état du réseau sur les coûts et les capacités de transit 156

niveau de R2 lorsque celle-ci intervenait dans la même relation que le niveau de R4 et avec un signe opposé. Nous obtenons alors pour le critère de convergence de la méthode de programmation linéaire successive les valeurs figurant dans le tableau 7.3.

Itération	1	2	3	4	5	6	7	8	9
Critère de convergence (m^3)	16524,2	2068,0	1384,1	511,0	1120,8	75,9	11,1	1,0	0,05

Tableau 7.3: Evolution du critère de convergence après suppression des influences multiples

Enfin, nous pouvons revenir sur l'étude des deux solutions obtenues au paragraphe 7.3.6.2 pour constater qu'elles aboutissent à des coûts qui diffèrent de moins de 1 ‰. Ceci signifie que même si il existe des conditions hydrauliques créant une non-convexité locale, il restera possible de retenir un des optima locaux.

7.4 Conclusion

L'écriture complète de la dépendance des transits vis-vis de l'état montre clairement la non-linéarité du problème. Cette non-linéarité ne peut pas être contournée par l'introduction de variables binaires, le problème restant à contraintes quadratiques avec des variables mixtes. Par contre, la résolution par programmation linéaire successive peut dans certains cas aboutir en un temps court à la convergence. En particulier nous avons mis en évidence la convergence sur notre site d'étude. Pour définir les contours du domaine de convergence de cette méthode, nous avons augmenté artificiellement l'influence des niveaux des réservoirs sur les débits circulant sur le réseau. L'augmentation effectuée est équivalente à une multiplication par dix des hauteurs d'eau dans les réservoirs. De plus, il est possible de définir des règles générales d'obtention de domaines d'admissibles localement convexes.

Conclusion

Les objectifs fixés initialement ont été atteints à des degrés divers.

Les modèles de prévision de la demande de l'heure suivante donnent de bons résultats puisque les coefficients de corrélation entre les prévisions et les données observées sont de l'ordre de 0.96. La prévision à plus long terme ne répond pas complètement à la volonté initiale. La dégradation est très nette à partir d'un horizon de 10 heures. L'idée de construire un modèle d'optimisation dont les calculs ne demanderaient à être effectués qu'au début d'un horizon de 24 heures semble encore irréalisable. Cette idée n'est d'ailleurs pas intéressante d'un point de vue pratique puisqu'il sera sans doute nécessaire de réajuster l'état du réseau en fonction de l'information recueillie par les différents capteurs.

L'erreur résiduelle de prévision est bien caractérisée. Trois méthodes de prise en compte de l'erreur sont proposées et appliquées à un réseau exemple. La méthode dite de "formulation serrée" permet de fortement sécuriser le système au détriment du coût de l'optimisation. Dans la formulation "probabilisée" chacune des contraintes d'un problème général se voit attribuer une probabilité de respect. Cela permet au modélisateur de choisir les réservoirs et les moments de la journée où le respect de la contrainte est primordial. Cette méthode permet plus de souplesse que la "formulation serrée" et amène donc à des coûts inférieurs. La formulation de la "programmation stochastique en deux étapes" est assez proche de la "formulation serrée" mais il s'agit cette fois d'attribuer des pénalités à des consommations qui s'avèreraient plus fortes que la moyenne. Le choix de pénalités peut être plus aisé à manipuler que celui de probabilités de respect. La méthode de résolution est complètement différente et l'on peut observer les nouvelles consommations attribuées par le modèle en fonction des pénalités. Les trois méthodes permettent de quantifier le risque pris par un modèle d'optimisation et permettront de sécuriser l'optimisation en fonction de probabilités ou de pénalités que se donneront les exploitants d'un réseau.

Nous avons pu proposer deux écritures différentes permettant de choisir parmi des solutions de mêmes coûts, l'une par la programmation par les graphes et la seconde par la programmation linéaire. Dans le deuxième cas, nous avons montré comment nous pouvons choisir un deuxième critère d'optimisation. Le deuxième critère amène à choisir parmi les solutions initialement de mêmes coûts la solution assurant le maximum de sécurité pour un réservoir donné. L'apport de cet outil devrait aussi permettre de rassurer les utilisateurs des outils d'optimisation que nous proposons.

La prise en compte de contraintes de continuité est dans un premier temps modélisée

sous la forme de graphes décomposables en sous-graphes. Dans un deuxième temps, cette contrainte est entièrement écrite sous la forme d'un programme en variables mixtes. La mise en oeuvre sur des contre-exemples type laisse quelques incertitudes quant aux temps de calcul et quant à la stabilité de ces temps de calcul. Sur des cas réels, il est possible de guider davantage les calculs en utilisant les résultats des calculs d'optimisation de la veille. Il sera nécessaire, par la suite, de continuer à travailler sur la maîtrise des temps de calcul en quantifiant les écarts de solutions optimales entre deux jours successifs.

La prise en compte de l'influence de l'état du réseau sur les transits entre ouvrages est elle-aussi proposée par deux méthodes différentes. Le premier modèle consiste en une écriture sous forme d'un programme linéaire en variables mixtes. L'Utilisation de variables booléennes permet de caractériser l'état de chacun des réservoirs grâce à une discrétisation des volumes. Le programme proposé n'étant pas linéaire, il n'est pas possible de le résoudre. La deuxième formulation repose sur la programmation linéaire successive. Elle donne des résultats pratiques probants puisque sur la partie ouest du réseau du Syndicat des Eaux de la Région de Sarralbe que nous avons modélisée, la convergence a lieu après quelques itérations. Il reste toutefois difficile d'établir la limite entre les configurations pour lesquelles la méthode converge et celles pour lesquelles la non-linéarité et la non-convexité posent des problèmes plus sérieux. Lorsqu'il n'y a pas convergence vers une solution unique, il est possible que les solutions multiples ne diffèrent que très peu les unes des autres permettant de retenir l'une d'elles comme optimum.

L'ensemble de la démarche menée dans ce travail se voulait transposable à divers types de réseaux. Aucune des étapes abordée ne semble contredire cette volonté initiale. Toutefois, seule une mise en oeuvre pratique sur des réseaux de configurations différentes permettra de vérifier que cet objectif a été atteint. Il sera notamment indispensable de rendre automatique les liaisons entre les sous-parties que nous avons présentée. Ainsi faudra-t-il que la prévision de la demande, le calcul des paramètres stochastiques et l'optimisation elle-même constitue un seul et même outil de gestion optimale des réseaux d'eau potable. Par ailleurs, il est indispensable pour que ces outils soient effectivement utilisés par les exploitants que l'utilisateur final ait la possibilité d'intervenir sur les choix proposés. Dans les méthodes que nous avons élaborées, il sera possible d'une part de lui donner une information pertinente, sur la variabilité de la demande à venir par exemple, mais aussi de lui donner la possibilité de fixer des paramètres comme le critère de choix entre plusieurs solutions de même coût ou encore les pénalités à appliquer en cas de consommations plus fortes que celles attendues.

Bibliographie

- [1] (Abdollahzadeh, 1997) ABDOLLAHZADEH, Ali Akbar. Validation de données par équilibrage de bilans, synthèse et nouvelles approches.-Institut National Polytechnique de Lorraine, octobre 1997. (Doctorat).
- [2] (Aleksandrov, 1984) ALEKSANDROV, I.A., BULATOV, V.P., OGNIVTSEV, S.B. et al. Solution of a stochastic programming problem concerning the distribution of water resources.-Stochastic Optimization. Proceedings of the International Conference, Kiev, 1984.
- [3] (Amoretti, 1990) AMORETTI, René. Modélisation et commande optimale d'un réseau de distribution d'eau potable. Mise en oeuvre, test et étude des performances sur le réseau du FIUM'ORBO.-Université Aix-Marseille III, mars 1990, 148 p. (Doctorat).
- [4] (Amoretti, 1996) AMORETTI, René, CARPITA, J.P. Modélisation et commande de réseau d'eau potable.-TSM, N° 3, janvier 1996, pp. 186-190.
- [5] (Anderson, 1989) ANDERSON, E.J., PHILPOTT, A.B. A continuous-time network simplex algorithm.-Networks, Vol.19, 1989, pp. 395-425.
- [6] (Baker, 1985) BAKER, Thomas E., LASDON, Léon S. Successive linear programming at Exxon.-Management Science, N° 3, Vol.31, March 1985, pp. 264-274.
- [7] (Berge, 1983) BERGE, C. Graphes.-Bordas, Gauthier-Villars, 1983, 400 p.
- [8] (Berghout, 1997) BERGHOUT, Brendan L., KUCZERA, George. Network linear programming as pipe network hydraulic analysis tool.-Journal of Hydraulic engineering-ASCE, N° 6, Vol.123, june 1997, pp. 549-559.
- [9] (Bigot, 1995) BIGOT, Pascal. Utilisation des réseaux de neurones pour la télégestion des réseaux techniques urbains.-Université de Lyon 1, 1995, 116 p. (Doctorat).
- [10] (Bissery, 1994) BISSERY, Christophe. La détection centralisée des fuites sur les réseaux d'eau potable par réseaux de neurones.-INSA Lyon, décembre 1994, 233 p. (Doctorat).
- [11] (Boniface, 1992) BONIFACE, Valérie. Logiciel d'optimisation économique du pompage "grands réseaux" de la Safege.-ENITRTS, mai 1992.
- [12] (Boucheron, 1995) BOUCHERON, Stéphane, CANU, Stéphane, GALLINARI, Patrick et al. De la représentation à la validation : la généralisation dans les réseaux de neurones.-Ecole Modulad - ASU, 23 p.

- [13] (Brémond, 1987) BREMOND, Bernard. Théorie et approche algorithmique du renforcement optimal des réseaux de transport d'eau.- Institut National Polytechnique de Toulouse, septembre 1987, 165 p. (Doctorat).
- [14] (Brémond, 1992) BREMOND, Bernard, GUINAUDEAU, Michel, MARCHAL, Denis. Gestion en temps réel du réseau d'eau potable de la ville de Bellegarde. Mise en oeuvre de la méthode et intérêt économique.-TSM, N° 5, mai 1992, pp. 245-251.
- [15] (Brion, 1991) BRION, L.M., MAYS, Larry W. Methodology for optimal operation of pumping stations in water distribution systems.-Journal of hydraulic engineering-ASCE, N° 11, Vol.117, 1991, pp. 1551-1569.
- [16] (Bryds, 1992) BRYDS, M.A. An algorithm for optimal scheduling of a class of cascade water supply systems.-Optimal control applications and methods, Vol.13, 1992, pp. 265-287.
- [17] (Canu, 1990) CANU, Stéphane, SOBRAL, Rul, LENGELLE, R. Formal neural networks as an adaptive model for water demand.-Proceedings of International Neural Networks Confence, Paris, 1990, pp. 131-135.
- [18] (Canu, 1995) CANU, Stéphane, DING, Xia, GRANDVALET, Yves. Réseaux de neurones pour la prévision à un pas de temps. Statistiques et méthodes neuronales. Support de cours.-Ecole Modulad - ASU, octobre 1995, 23 p.
- [19] (Canu, 1996) CANU, Stéphane, GRANDVALET, Yves. Les réseaux de neurones artificiels : estimation de dépendances à partir de données empiriques.-Université de Technologie de Compiègne, Support de cours, 1996.
- [20] (Carpentier, 1983) CARPENTIER, Pierre. Décomposition et agrégation dans la conduite optimale d'un grand réseau de distribution d'eau.-ENSMP, décembre 1983, 132 p. (Doctorat).
- [21] (Carpentier, 1993) CARPENTIER, Pierre, COHEN, Guy. Applied-Mathematics in Water-Supply Network Management.-Automatica, N° 5, Vol.29, 1993, pp. 1215-1250.
- [22] (Cembrano, 1988) CEMBRANO, G., BRYDS, M.A., QUEVEDO, J. et al. Optimization of a multi-reservoir water network using a conjugate gradient technique. A case study.-8th international conference on analysis and optimization of systems, Antibes, june 1988, pp. 987-999.
- [23] (Charrat, 1995) CHARRAT, Michel. Conception d'un outil de supervision de la production et de la distribution d'eau potable à Lyon en période courante et en période de crise.-INSA Lyon, mai 1995, 278 p. (Doctorat).
- [24] (Chatenet-Laurent, 1997) CHATENET-LAURENT, Nathalie. Algorithmes d'apprentissage de la politique optimale d'un processus stochastique. Application à un réseau d'alimentation en eau potable.-Université de Bordeaux I, juin 1997, 121 p. (Doctorat).
- [25] (Chvatal, 1983) CHVATAL, V. Linear programming.-Freeman and Co, 1983, 380 p.

- [26] (Ciampi, 1995) CIAMPI, Antonio, LECHEVALIER, Yves. Reseaux de neurones et modèles statistiques.-La Revue de Modulad, N° 15, 1995, pp. 27-46.
- [27] (Coutrot, 1984) COUTROT, Bernard, DROESBEKE, Fernand. Les méthodes de prévision.-Presses Universitaires de France col. Que sais-je?, 1984, 127 p.
- [28] (Crawley, 1993) CRAWLEY, Philip D., DANDY, Graeme C. Optimal operation of multiple-reservoir system.-Journal of Water Resources Planning and Management, N° 1, Vol.119, Februar, 1993, pp. 1-17.
- [29] (Crommelynck, 1991) CROMMELYNK, V., DUQUESNE, C., MERCIER, M. et al. Des outils de prévision journalière et horaire de la consommation d'eau par réseaux de neurones.-71^{ème} congrès de l'AGHTM. Annecy, avril 1991, pp. 429-443.
- [30] (Darnault, 1991) DARNAULT, Perre-Henri. Réalisation d'un système expert d'aide à la reconstitution d'un réseau électrique de transport après un incident généralisé.-Institut National Polytechnique de Grenoble, janvier 1991, 113 p. +ann. (Doctorat).
- [31] (David, 1989) DAVID, Marc, MICHAUD, Jean-Claude. La prévision : approche empirique d'une méthode statistique.-Masson, 1989, 187 p.
- [32] (Ding, 1995) DING, Xia, CANU, Stéphane, DENOEU, T. Neural network based models for forecasting.-Conférence ADT'95, juin 1995, 11 p.
- [33] (Du, 1997) DU, Yang-Guang, HODOUIN, Daniel, THIBAUT, Jules. Use of a novel autoassociative neural network for nonlinear steady-state data reconciliation.-AIChE Journal, N° 7, Vol.43, Juillet 1997.
- [34] (Dziegielewski, 1993a) DZIEGIELEWSKI, Benedykt, STRUS, Craig A., OPITZ, Eva M. Water demand management planning with IWR-Main Version 6.0.-Proceedings of conserv 93; The new water agenda, Las Vegas, 1993, pp. 1223-1228.
- [35] (Dziegielewski, 1993b) DZIEGIELEWSKI, Benedykt, STRUS, Craig A. Managing urban water demands.-1993 annual conference proceedings; American Water Works Association, San Antonio, 1993, pp. 243-253.
- [36] (Eremin, 1984) EREMIN, I.I., VATOLIN, A.A. Duality in improper mathematical programming problems under uncertainty.-Stochastic Optimization. Proceedings of the International Conference, Kiev, 1984, pp. 326-333.
- [37] (Ferris, 1990) FERRIS, M. C. Iterative linear programming solution of convex programs.-Journal of optimization theory and applications, N° 1, Vol.65, April, 1990, pp. 53-65.
- [38] (Ferstler, 1998) FERSTLER Vincent. Gestion en temps réel des réseaux d'eau potable. Prévision de consommation et calculs des capacités de transit entre ouvrages-ENGES, juillet 1998, 108 p.
- [39] (Flowers, 1991) FLOWERS, Greeg W., BAYLES, John, JAMIESON, R.A. Computer assisted high service pump selection.-Proceedings AWWA seminar on computers in the water industry, Austin, 1991, pp. 751-758.

- [40] (Gauriau, 1996) GAURIAU, G., DEVAUCHELLE, V. La maîtrise des réseaux de distribution d'eau potable. Valorisation avancée des débits acquis en continu.-TSM, N° 2, février 1996, pp. 121-126.
- [41] (Gilbert, 1988) GILBERT, Denis. Gestion optimisée des coûts de pompage. Etude du syndicat de Sarralbe.-Université Louis Pasteur de Strasbourg-ENITRTS, juin 1988, 81 p.
- [42] (Gilbert, 1990) GILBERT, Denis. Gestion automatisée en temps réel des réseaux d'eau potable. Calcul des commandes par différentes approches.-Université Louis Pasteur de Strasbourg-ENITRTS-Cemagref, octobre 1990, 132 p. (Doctorat).
- [43] (Gilbert, 1994) GILBERT, Denis, PEROT, M., OUASSI, Anas. Gestion optimale en temps réel d'un réseau AEP.-HYDROTOP, 12-15 avril 1994, juin 1994, pp. 113-121.
- [44] (Golstein, 1973) GOLSTEIN, E., YODINE, D. Problèmes particuliers de la programmation linéaire.-Editions Mir, 1973.
- [45] (Gondran, 1985) GONDRAN, Michel, MINOUX, Michel. Graphes et algorithmes.-Direction des Etudes et Recherches d'Electricité de France, 1985, 545 p.
- [46] (Gourieroux, 1995) GOURIEROUX, Christian, MONFORT, Alain. Séries temporelles et modèles dynamiques.-Economica, 1995, 664 p.
- [47] (Gravel, 1996) GRAVEL, Serge. Modélisation mathématique de la variation du niveau des réservoirs à l'usine de Sainte-Foy et au poste de pompage Plateau.-Vecteur Environnement, N° 2, Vol.29, avril 1996, pp. 73-78.
- [48] (Grombach, 1982) GROMBACH, P. Economic models based on utilization of linear programming to select the most economical solution in providing water.-Congrès AIDE 1982 Zürich, 1982, pp. 18-25.
- [49] (Guhl, 1995) GUHL, Florent. Gestion automatisée des réseaux d'eau potable. Prise en compte de contraintes de continuité.-Université Louis Pasteur de Strasbourg-ENGES, Septembre 1995, 121 p.
- [50] (Guhl, 1998) GUHL, Florent, BREMOND, Bernard, GILBERT, Denis. Drinking water demand using an hourly timestep. 5th International Conference on Neural Information Processing, Kitakyushu, October 1998
- [51] (Guillon, 1991) GUILLON, M., CROMMELYNK, V. Les applications de l'intelligence artificielle à la gestion des réseaux de distribution.-TSM, N° 11, novembre 1991, pp. 519-526.
- [52] (Guinaudeau, 1989) GUINAUDEAU, Michel, GILBERT, Denis, WACHEUX, H. Gestion en temps réel d'un réseau d'eau potable.-TSM, N° 9, septembre 1989, pp. 459-468.
- [53] (Hillier, 1980) HILLIER, Frederick S., LIEBERMAN, Gerald J. Introduction to operations research.-Holden-Day, Inc., 1980, 829 p.

- [54] (Jain, 1994) JAIN, Ashu, ORMSBEE, Lindell E. Short-Term municipal water use forecast modeling using artificial intelligence.-Computers in the water industry, Los Angeles, april 1994.
- [55] (Kim, 1985) KIM, S.W., LEE, S.T., JI, H.K. Forecasting models for future urban water demands.-5th Asia Pacific Regional Water Supply Conference, September 1985.
- [56] (Kuczera, 1989) KUCZERA, George. Fast multireservoir multiperiod linear programming models.-Water Ressources Research, N° 2, Vol.25, Februar 1989, pp. 169-176.
- [57] (Kuczera, 1992) KUCZERA, George. Water supply headworks simulation using network linear programming.-Advances in engineering software, N° 14, 1992, pp. 55-60.
- [58] (Kuczera, 1993) KUCZERA, George. Network linear programming codes for water-supply headworks modeling.-Journal of Water Resources Planning and Management, N° 3, Vol.119, June 1993, pp. 412-417.
- [59] (Lannuzel, 1990) LANNUZEL, Pierre. Recueil et intégration d'heuristiques dans les algorithmes de définition de stratégies de pompage.-Ecole Nationale des Ponts et Chaussées, janvier 1990, 183 p. + ann. (Doctorat).
- [60] (Lansey, 1994) LANSEY, Kevin E., AWUMAH, Kofi. Optimal pump operations considering pump switches.-Journal of Water Resources Planning and Management, N° 1, Vol.120, January 1994, pp. 17-35.
- [61] (Lek, 1996) LEK, S., DIMOPOULOS, I., DERRAZ, M. et al. Modélisation de la transformation Pluie-Débit en utilisant les réseaux de neurones artificiels.-Revue des sciences de l'eau, N° 3, Vol.9, 1996.
- [62] (Little, 1991) LITTLE, Keith W., MOREAU, David H. Estimating the effects of conservation on demand during droughts.-Journal American Water Works Association, N° 10, Vol.83, October 1991, pp. 48-54.
- [63] (Lobrecht, 1997) LOBBRECHT, Arnold H. Dynamic Water-System Control. Design and Operation of Regional Water-Ressources Systems.-A.A.Balkema, 1997, 315 p.
- [64] (Mackle, 1995) MACKLE, Gunther, SAVIC, Dragan A., WALTERS, Godfrey A. Application of Genetic Algorithms to Pump Scheduling for Water Supply.-Genetic Algorithms in Engineering Systems : Innovations and Applications, GALESIA '95, IEE , N° 414, 1995, pp. 400-405.
- [65] (Mangeas, 1993) MANGEAS, Morgan, COTTRELL, Marie, GIRARD, Bernard et al. Advantages of multilayer perceptron for modeling and forecasting time series : application to the daily electrical consumption in France.-EDF, Direction des Etudes et Recherches, novembre 1993, 13 p.
- [66] (Matas, 1996) MATAS, Christophe. Système expert de supervision du réseau d'eau potable.-Gas, Wasser, Abwasser, mai 1996, pp. 399-402.

- [67] (Mészáros, 1996) MESZAROS, Csaba. The efficient implementation of interior point methods for linear programming and their applications.-Eotvos Lorand University of Sciences, Budapest, 1996, 54 p. (PhD).
- [68] (Meyer, 1990) MEYER, Michel. Validation de données sur des systèmes incomplètement observés.-Institut National Polytechnique de Toulouse, juillet 1990. (Doctorat).
- [69] (Minoux, 1983a) MINOUX, Michel. Programmation mathématique. Théorie et algorithmes. Tome 1.-Dunod, 1983, 294 p.
- [70] (Minoux, 1983b) MINOUX, Michel. Programmation mathématique. Théorie et algorithmes. Tome 2.-Dunod, 1983, 229 p.
- [71] (Muller, 1993) MULLER, Corinne, MANGEAS, Morgan. Réseaux de neurones et prévision de séries temporelles.-EDF, Direction des Etudes et Recherches, 1993, 77 p.
- [72] (Muller, 1994) MANGEAS, Morgan, MULLER, Corinne, COTTRELL, Marie et al. A neural network tool for forecasting french electricity consumption.-EDF, Direction des Etudes et Recherches, September 1994, 10 p.
- [73] (Murat, 1997) MURAT, Cécile. Les problèmes d'optimisation combinatoire probabilistes dans les graphes.-Université de Paris-Dauphine, septembre 1997, 188 p. (Doctorat).
- [74] (Nobert, 1995) NOBERT, Yves, OUELLET, Roch, PARENT, Régis. La recherche opérationnelle.-Gaëtan Morin, 1995, 450 p.
- [75] (Ormsbee, 1994) ORMSBEE, Lindell E., LANSEY, Kevin E. Optimal control of water supply pumping systems.-Journal of Water Resources Planning and Management, N° 2, Vol.120, April 1994, pp. 237-252.
- [76] (Ouassi, 1993) OUASSI, Anas. Validation d'un logiciel de gestion optimisée des coûts de pompage. Etude sur le Syndicat de Sarralbe.-ENSEEIH, 1993, 26 p.
- [77] (Palacios-Gomez, 1982) PALACIOS-GOMEZ, F., LASDON, Léon S., ENGLISH, M. Nonlinear optimization by successive linear programming.-Management Science, N° 28, Vol.10, October 1982, pp. 1106-1120.
- [78] (Papon, 1994) PAPON, Max. The link between meteorology and electrical consumption : the usefulness of weather forecasts.-EDF, Direction des Etudes et Recherches, 1994, 15 p.
- [79] (Pellegrini, 1995) PELLEGRINI, Roger, COURTOIS, Philippe. Contrôle-commande. Voyage au coeur de la Sagep.-Environnement et Technique, N° 148, juillet 1995, pp. 38-40.
- [80] (Peng, 1995) PENG, Yibin. Méthodes d'optimisation et d'aide à la décision pour l'exploitation et l'extension de réseaux de chauffage à distance.-Ecole Polytechnique Fédérale de Lausanne, 1995, 208 p. (Doctorat).

- [81] (Pez, 1996) PEZ, Georgette, JARRIGE, Pierre-Antoine, TOCQUEVILLE, Laurent et al. Un modèle hydraulique de réseau d'eau potable en temps réel.-L'eau, l'industrie, les nuisances, N° 196, novembre 1996, pp. 49-53.
- [82] (Pezeshk, 1994) PEZESHK, S, HELWEG, Otto J., OLIVIER, K.E. Optimal operation of ground-water supply distribution systems.-Journal of Water Resources Planning and Management, N° 5, Vol.120, October 1994, pp. 573-586.
- [83] (Ragot, 1993) RAGOT, José, MAQUIN, Didier, GADDOUNA, Besma. Supervision des flux de production dans les processus continus.-4^{ème} Congrès de la Minéralurgie, Grenoble 1993, Octobre 1993, 17 p.
- [84] (Rakotondrabe, 1988) RAKOTONDRABE, Herintsoavolona. Analyse des enregistrements de débits à pas de temps fin sur le réseau du District Urbain de Nancy. Méthodologie de détection d'anomalies et de validation de données.-ENITRTS, juin 1988, 91 p. +ann.
- [85] (Ramirez-Beltran, 1993) RAMIREZ-BELTRAN, N.D., JIMENEZ-CEDENO, M.A., ALLISON, J.T. Optimization of electrical energy distribution for a deficient system.-Journal of the operationnal research society, N° 8, Vol.44, 1993, pp. 761-772.
- [86] (Renaud, 1993) RENAUD, Arnaud. Daily generation management at Electricité de France : From planning towards real time.-IEEE Transactions on automatic control, N° 7, Vol.38, July 1993, pp. 1080-1093.
- [87] (Rhoades, 1995) RHOADES, Stephen D. Hourly monitoring of single-family residential areas.-Journal American Water Works Association, N° 8, Vol.87, 1995, pp. 43-49.
- [88] (Ritzel, 1994) RITZEL, Brian J., RANJITHAN, S., WAYLAND EHEART, J. Using genetic algorithms to solve a multiple objective groundwater pollution containment problem.-Water Ressources Research, N° 5, Vol.30, mai 1994, pp. 1589-1603.
- [89] (Rothstein, 1991) ROTHSTEIN, Eric. The city of AUSTIN, Texas' hourly water demand monitoring project.-1991 annual conference proceedings; American Water Works Association, Philadelphia, 1991, pp. 651-671.
- [90] (Rothstein, 1993) ROTHSTEIN, Eric. Summer water demand patterns in a southwest city.-Proceedings of conserv 93; The new water agenda, Las Vegas, 1993, pp. 1491-1516.
- [91] (Sakarovitch, 1984a) SAKAROVITCH, Michel. Optimisation combinatoire-Graphes et programmation linéaire.-Hermann, 1984, 249 p.
- [92] (Sakarovitch, 1984b) SAKAROVITCH, Michel. Optimisation combinatoire- Programmation discrète.-Hermann, 1984, 269 p.
- [93] (Savic, 1995) SAVIC, Dragan A., WALTERS, Godfrey A. Place of Evolution Programs in Pipe Network Optimization.-22nd Annual Conference of the ASCE Water Resources Planing and Management Division, May 7-, 1995, 4 p.

- [94] (Savic, 1996) SAVIC, Dragan A., WALTERS, Godfrey A. Stochastic Optimization Techniques in Hydraulic Engineering and Management.-7th International Symposium on Stochastic Hydraulics, Mackay, Australia, July, 1996, 8 p.
- [95] (Savic, 1997) SAVIC, Dragan A., WALTERS, Godfrey A., SCHWAB, Martin. Multiobjective Genetic Algorithms for Pump Scheduling in Water Supply.-Evolutionary Computing workshop, AISB '97, Manchester, 7-8 April 1997, 9 p.
- [96] (Shamir, 1974) SHAMIR, Uri. Optimal design and operation of water distribution systems.-Water Resources Research, N° 1, Vol.10, Februar 1974, pp. 27-36.
- [97] (Shamir, 1979) SHAMIR, Uri. Optimization in water distribution systems engineering.-Mathematical programming study, Februar 1979, pp. 65-84.
- [98] (Sobral, 1991) SOBRAL, Rul, CANU, Stéphane. Application des réseaux de neurones artificiels à la prévision : la consommation d'eau.-71^{ème} congrès de l'AGHTM. Annecy, avril 1991, pp. 399-409.
- [99] (Sterling, 1975) STERLING, M.J.H., COULBECK, B. Optimization of water pumping costs by hierarchical methods.-Proc. Instn Civ. Engrs, Part 2, December 1975, pp. 789-797.
- [100] (Ulanicki, 1991) ULANICKI, B., ORR, C.H. Unified approach for the optimization of nonlinear hydraulic systems.-Journal of optimization theory and applications, N° 1, Vol.68, 1991, pp. 161-179.
- [101] (Ulanicki, 1993) ULANICKI, B., RANCE, J.P., DAVIS, D. et al. Computer-aided optimal pump selection for water distribution networks.-Journal of Water Resources Planning and Management, N° 5, Vol.119, October 1993, pp. 542-562.
- [102] (Wonnacott, 1984) WONNACOTT, Ronald J., WONNACOTT, Thomas H. Statistique, économie, gestion, science, médecine.-Economica, 4^{ème} éd. 1991.
- [103] (Wu, 1995) WU, Xuan, ZHOU, Xiaoxin, HU, Zhaoyi. The recursive forecasting models with neural network and its use in forecasting.-Power System Technology, N° 12, Vol.19, December 1995, pp. 11-15.
- [104] (Yu, 1994) YU, G., POWELL, R.S., STERLING, M.J.H. Optimized pump scheduling in water distribution systems.-Journal of optimization theory and applications, N° 3, Vol.83, December 1994, pp. 463-488.
- [105] (Zessler, 1989) ZESSLER, U., SHAMIR, Uri. Optimal operation of water distribution systems.-Journal of Water Resources Planning and Management, N° 6, Vol.115, November 1989, pp. 735-752.