

HEC MONTRÉAL

**Contrôle optimal d'options réelles par apprentissage par renforcement :
application à la gestion hydroélectrique.**

par

Maël Véron

**Sciences de la gestion
(Option Ingénierie financière)**

*Mémoire présenté en vue de l'obtention
du grade de maîtrise ès sciences
(M. Sc.)*

Avril 2019
© Maël Véron, 2019

Résumé

La Canada dispose de vastes ressources hydroélectriques qui le place parmi les premiers producteurs mondiaux. De ce fait, son avenir énergétique dépend autant de la préservation de ses ressources que de sa capacité à optimiser la gestion de ses centrales hydro-électriques par des méthodes de plus en plus performantes. L'objet de ce mémoire s'attache à ce dernier enjeu et propose d'expérimenter une méthode d'aide à la décision appliquée à la gestion hydroélectrique : la recherche de politique directe. L'objectif est double : évaluer la capacité d'un réseau de neurones à améliorer la production de mégawatts puis comparer la performance de cet outil par rapport aux méthodes de programmation dynamique actuellement utilisées par le système Kemano de l'entreprise Rio Tinto situé en Colombie-Britannique. Plus spécifiquement, la performance de ces deux outils d'aide à la gestion électrique sera évaluée en fonction de deux paramètres : l'optimisation sous contraintes des mégawatts et la réduction de la variabilité des résultats à l'approche de notre optimum. Les mégawatts moyens générés à chaque année sont équivalents à ce qui est obtenu à l'aide des méthodes actuelles.

Mots-Clés

Système hydroélectrique, gestion de réservoirs, apprentissage par renforcement, réseaux de neurones

Abstract

Numerous hydroelectric resources are available in Canada, placing it among the world's leading producers. As a result, its energy future depends as much on the preservation of its resources as on its ability to optimize the management of its hydroelectric plants by increasingly efficient methods. Thus, the subject of this paper focuses on the latter issue and proposes to experiment with a method of decision support applied to hydroelectric management : direct policy research. The goal of our research is twofold : to evaluate the ability of a neural network to improve the megawatt production, and then compare the performance of this tool against the dynamic programming methods currently used by Rio Tinto's Kemano system, located in British Columbia. More particularly, the performance of these two power management tools will be evaluated according to two criteria : the optimization under constraints of the megawatt production and the reduction of the results' variability when approaching our optimum. The average megawatts production generated each year are equivalent to what is achieved using current methods.

Keywords

hydroelectric system, tank management, reinforcement learning, neural networks.

Table des matières

Résumé	i
Abstract	iii
Table des matières	iv
Liste des tableaux	vii
Liste des figures	ix
Remerciements	xi
1 Introduction	1
1.1 Objectifs de recherche	1
1.2 Revue de littérature	2
1.3 Structure du mémoire	4
2 Description d'un système hydroélectrique : le cas de Kemano	7
2.1 Description du système hydroélectrique de Kemano	7
2.2 Modélisation mathématique d'un système hydroélectrique	10
2.2.1 Contraintes et optimisations du système de Kemano	10
3 Résolution du problème de gestion hydroélectrique	17
3.1 Algorithme de résolution et optimisation	17
3.2 Réseau de neurones en gestion hydroélectrique	18

3.2.1	Introduction aux réseaux de neurones	18
3.2.2	Utilisation des réseaux de neurones comme politique de gestion	22
4	Mise en oeuvre et résultats numériques	27
4.1	Contraintes de fonctionnement et mégawatts	28
4.2	Décisions et méthodes d'optimisations	31
4.2.1	Choix des débits vers Skins Lake et Kemano	31
4.3	Méthodes d'optimisation	35
4.3.1	Méthode 1	35
4.3.2	Méthode 2	36
4.3.3	Méthode 3	37
4.3.4	Méthode 4	38
4.3.5	Méthode 5	39
4.3.6	Méthode 6	40
4.3.7	Méthode 7	40
4.4	Interprétations des résultats et remarques supplémentaires.	42
4.5	Changement du nombre d'actions disponibles et des pénalités d'inon- dations	45
4.6	Tests de sensibilité et tests supplémentaires	46
4.6.1	Utilisation de poids aléatoires	48
4.7	Observations sur les apports et la politique	53
4.7.1	Variation des apports simulés	53
4.7.2	Variation des actions utilisées par le réseau de neurones lors des phases d'apports élevés	54
	Conclusion	57
	Bibliographie	59

Liste des tableaux

2.1	Tableau des différentes variables et de leur unité	15
3.1	Matrice des données utilisées pour l'optimisation de la politique	18
4.1	Décisions possibles pour le système hydroélectrique de Kemano	32
4.2	Inondations moyennes pour la méthode 3	38
4.3	Probabilités d'inondations pour la méthode 5	39
4.4	Actions modifiées du système hydroélectrique de Kemano	45
4.5	Pourcentage moyen des mégawatts en fonction de la puissance maximale atteignable pour les meilleurs poids obtenus avec Nelder-Mead grâce à la méthode 7 et les meilleurs poids aléatoires optimisés par Nelder-Mead	50
4.6	Pourcentage d'années avec une production de mégawatts inférieure à 700 pour les meilleurs poids obtenus avec Nelder-Mead en utilisant la méthode 7 et les meilleurs poids aléatoires optimisés par Nelder-Mead	50
4.7	Pourcentage d'années en inondations pour les meilleurs poids obtenus avec Nelder-Mead en utilisant la méthode 7 et les meilleurs poids aléatoires optimisés par Nelder-Mead	50
4.8	Pourcentage d'années lors desquelles le réservoir atteint sa limite inférieure pour les meilleurs poids obtenus avec Nelder-Mead en utilisant la méthode 7 et les meilleurs poids aléatoires optimisés par Nelder-Mead	51

Liste des figures

2.1	Apports simulés pour 5 années	8
2.2	Plan du système hydroélectrique de Kemano	9
2.3	Schéma du système hydroélectrique de Kemano	11
2.4	Graphique des limites maximales et minimales selon les périodes	12
2.5	Fonction de conversion des élévations en volume	13
3.1	Réseau de neurone avec 3 unités d'entrées, 2 unités de sorties et une couche cachée de 3 neurones	20
3.2	Schéma de l'algorithme complet d'optimisation	26
4.1	Élévations régulières pour 5 scénarios pour la méthode 7	41
4.2	Élévations irrégulières pour 5 scénarios pour la méthode 7	42
4.3	Mégawatts réguliers pour la méthode 7 et simulation 1	43
4.4	Mégawatts irréguliers pour la méthode 7 et simulation 1	43
4.5	Mégawatts moyen par année pour la méthode 7 et simulation 1	44
4.6	Mégawatts moyen par année pour la méthode 7 et simulation 1	44
4.7	Évolution de la valeur de la fonction objectif pour les 10 000 premières itérations de poids aléatoires	49
4.8	Mégawatts moyen en pourcentage de la capacité atteignable par années avec l'ensemble d'apport sim 1 avec les poids obtenus de Nelder-Mead en partant de poids aléatoires	51

4.9	Mégawatts moyen en pourcentage de la capacité atteignable par années avec les apports historiques avec les poids obtenus de Nelder-Mead en partant de poids aléatoires	52
4.10	Nombre de scénarios avec l'apport naturel maximum pour les 20 premières périodes	53
4.11	Apports naturels pour plusieurs scénarios	54

Remerciements

Je remercie toutes les personnes m'ayant apporté leur aide lors de mon mémoire.

Merci à Michel Denault pour son soutien indéfectible, son temps et son aide tout au long du processus. Merci aussi à Pascal Côté de Rio Tinto pour son accueil au Saguenay. Ses connaissances du système hydroélectrique de Kemano et de l'utilisation de Python m'ont été d'une aide précieuse. Merci aussi à elyse pour son soutien moral depuis toutes ces années, ainsi qu'à Ori pour sa précieuse collaboration lors de la rédaction et la correction de mon mémoire. Merci à Sandrine pour avoir été à mes côtés depuis le début et pour avoir toujours cru en moi. Enfin, je remercie IVADO pour le soutien financier.

Chapitre 1

Introduction

Dans ce mémoire, il est question d'utiliser une approche de recherche de politique directe afin d'optimiser le système hydroélectrique de Kemano. Ce dernier se situe en Colombie-Britannique et appartient à la compagnie Rio Tinto. Elle fait partie des plus grands producteurs d'aluminium en Amérique du Nord et a donc besoin pour sa production d'une très grande quantité d'électricité. La centrale hydroélectrique de Kemano fournit l'électricité à l'aluminerie de Kitimat, et le surplus est revendu à la compagnie d'électricité provinciale BC Hydro. Il faut ajouter à cela des contraintes environnementales et d'autres liées à la sécurité du public. Enfin, comme pour tous les systèmes hydroélectriques, une grande part d'aléatoire entre en jeu lorsque l'on s'intéresse aux apports d'eau intervenant dans le remplissage des réservoirs et des lacs. Kemano ne fait pas exception à cette règle, et l'administrateur de cette usine doit ajuster ses décisions à ces approvisionnements stochastiques tout au long de l'année.

1.1 Objectifs de recherche

En règle générale, tout comme pour l'usine Kemano, la gestion des barrages hydroélectriques se focalise sur le court et le long terme. En effet, les décisions se doivent de convenir à la période actuelle sans mettre en péril l'approvisionnement

énergétique futur et les risques d'inondations à long terme. Le problème provient du fait que les variables en lien avec cette gestion contiennent une grande part d'aléatoire. Il existe déjà des méthodes d'optimisation qui tentent de palier le problème des apports aléatoires, dont les plus utilisées sont des algorithmes de programmation dynamique. L'objectif étant de permettre à l'administrateur de prendre les bonnes décisions pour chaque période, nous proposons l'utilisation de deux outils mathématiques tels qu'un réseau de neurones et un algorithme de Nelder-Mead. La recherche de politique directe permettra d'obtenir les politiques d'utilisation des ressources hydriques, et l'algorithme de Nelder-Mead contribuera à l'optimisation des paramètres de la politique. Nous nous servons des apports d'eaux simulés fournis par Rio Tinto. Nous nous attarderons aussi sur différents ajustements possibles du réseau afin d'améliorer les résultats.

1.2 Revue de littérature

Les méthodes les plus utilisées dans la gestion des réservoirs hydroélectriques sont la programmation linéaire (PL) et la programmation dynamique (PD). Une revue de ces différentes méthodes est donnée par Rani et Moreira (2009). Pour le sujet qui nous intéresse, c'est-à-dire la gestion des systèmes hydroélectriques, la PL trouve surtout son intérêt dans les problèmes à grande échelle, i.e. les barrages possédant de nombreux réservoirs. En revanche, il est nécessaire d'utiliser des fonctions objectives linéaires et convexe ainsi que des contraintes linéaires. On peut tout de même trouver des solutions à ces contraintes, comme le font Barros et al. (2003) à l'aide de PL successives. La PD est, sans contredit, l'approche la plus populaire pour résoudre les problèmes de gestion. Cette méthode a été introduite par Bellman (1957). Young (1967) et Hall (1968) feront partie des premiers auteurs à travailler sur le sujet de la gestion optimale des réservoirs. Le problème reste qu'ils utilisent la PD déterministe, ce qui force des approximations et ne correspond pas à la réalité stochastique des apports d'eau naturels. En ce qui concerne la problématique

des réservoirs, l'incertitude des apports est alors considérée dans la programmation dynamique stochastique (PDS). Stedinger (1984) a amélioré le processus en développant un modèle de PDS qui utilise la meilleure prévision comme une variable état plutôt que les apports. Pour améliorer cet algorithme, on peut utiliser la programmation dynamique stochastique par échantillonnage (SSDP), comme proposé par Kelman (1990). Avec cette dernière, on utilise directement les scénarios pour estimer la fonction valeur future. Malgré ces améliorations, la difficulté majeure est la taille du problème qui peut vite augmenter les temps de calcul, ce qui oblige à réaliser d'importantes approximations. La prochaine étape consisterait à développer une nouvelle méthode permettant de limiter ces approximations, tout en conservant, au maximum, la taille de notre problème.

Une nouvelle approche de la gestion des réservoirs va permettre, en plus d'améliorer les résultats, de tenter de lutter contre ce fléau. Elle se situe au niveau de l'apprentissage par renforcement. Cette dernière technique permet à l'algorithme d'apprendre de ses décisions mais aussi d'obtenir un résultat quantitatif (nommé récompense) lui permettant d'ajuster ses paramètres afin d'améliorer le résultat précédent. Plus simplement, un agent effectue une action, et reçoit en retour un nouvel état et une récompense. Il ajuste alors sa prochaine action de façon à améliorer la dite récompense.

Lee et Labadie (2007) utilise une telle méthode, nommée Reinforcement learning, dans le but de maximiser les récompenses à long terme. L'algorithme apprend à prendre de bonnes décisions directement en simulant le système. Il n'est donc pas nécessaire de les connaître à l'avance. Le processus utilise la fonction Q de Bellman, qui prend en compte l'action, l'état et les apports, et la compare avec la fonction de valeur. On peut ainsi laisser la procédure se poursuivre jusqu'à convergence, ou jusqu'à l'absence d'amélioration, afin de trouver les paires actions-états maximisant la récompense.

Les dernières techniques d'apprentissage par renforcement se portent sur l'utilisation des réseaux de neurones artificiels (RNA). Ils permettent de déterminer la

sortie correspondante à une certaine entrée après avoir été entraînés à l'aide d'un ensemble d'entrées-sorties. La première version, nommée perceptron, est proposée par Rosenblatt(1957) avec une simple couche. Ensuite, des versions plus complexes se succédèrent, notamment des perceptrons multicouches, permettant la résolution de problèmes plus complexes, comme la reconnaissance d'image, la reconnaissance facial voir même l'apprentissage de stratégies dans des jeux tels que les échecs ou le Go. Pour l'application au domaine de la gestion hydro-électrique, et dans le but de trouver une politique optimale, Cancelliere (2002) utilise un mélange de programmation dynamique et de réseaux de neurones. La PDS est d'abord utilisée pour déterminer la libération d'eau idéale. Ces résultats sont utilisés avec des RNA pour améliorer la performance dans des périodes de sécheresse ou d'inondations. Dans ce cas, c'est le réseau qui apprend directement, ce qui est différent de notre méthode, qui sera expliquée plus loin. Enfin, beaucoup plus récemment, Desreumaux (2016) cherche à trouver la meilleure politique de gestion hydro-électrique à l'aide d'un réseau de neurones. Il désire trouver la politique avec une descente de gradient, à l'aide de la méthode acteur-critique et des RNA. Sa principale contribution consiste en l'utilisation de la solution de référence obtenue grâce à la contrepartie déterministe de la politique stochastique afin de réduire la variance de l'estimation du gradient. Cela permet ainsi de converger beaucoup plus rapidement et d'obtenir une politique plus performante. Nous poursuivons dans cette direction en cherchant une politique optimale de gestion hydroélectrique faisant appel à un réseau de neurones. Ce dernier devra mettre à jour les variables à l'aide d'un algorithme de Nelder-Mead ajouté à la méthode.

1.3 Structure du mémoire

Le chapitre 2 propose une description du système hydroélectrique de Kemano ainsi qu'une modélisation mathématique du problème de gestion. Le troisième chapitre sera consacré d'une part à la présentation de l'algorithme de résolution et

d'autre part, à une introduction sur les réseaux de neurones qui serviront dans la résolution de notre problème d'optimisation des mégawatts liés à la production d'électricité. Enfin, le dernier chapitre fournira tous les détails sur la mise en oeuvre de notre algorithme et présentera les résultats numériques en vue des conclusions adéquates.

Chapitre 2

Description d'un système hydroélectrique : le cas de Kemano

Ce chapitre est divisé en 2 sections. Dans la première, nous décrivons le fonctionnement général d'un système hydroélectrique avec réservoir, en particulier celui de Kemano. On ajoute à cela une description de tous ses éléments. Dans la section suivante sera développée la modélisation mathématique du système.

2.1 Description du système hydroélectrique de Kemano

En règle générale, un système hydroélectrique est composé d'au moins 2 éléments principaux. Premièrement, les réservoirs, qui emmagasinent l'eau qui sera ensuite envoyée vers le deuxième élément en aval, les turbines. Ces dernières vont, grâce à la force du courant, tourner et produire de l'électricité. Les réservoirs se remplissent à l'aide des apports d'eau naturels, comme les précipitations et la neige fondante, et se vident lors de la libération en direction des turbines. À la vue des conditions permettant le remplissage, on comprend la nature aléatoire et saisonnière des apports. La figure 2.1 présente quelques apports et fournit un aperçu de leur caractère

aléatoire et saisonnier.

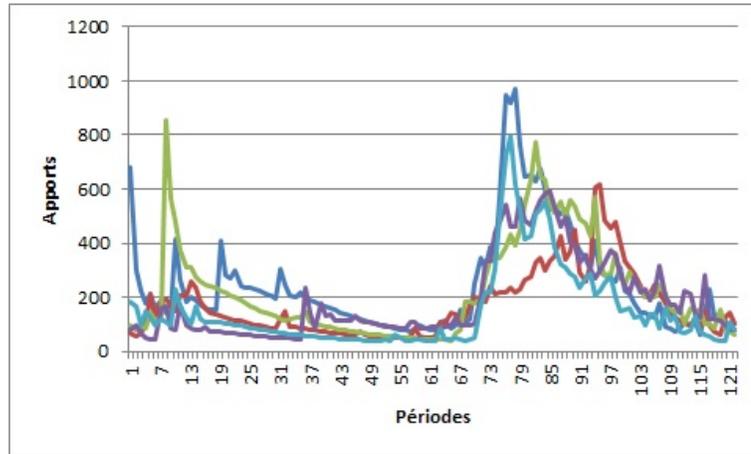


FIGURE 2.1 – Apports simulés pour 5 années

Le principal problème est alors de gérer correctement les décisions liées à l'évacuation de l'eau. En effet, une politique ayant une préférence pour une évacuation élevée risque de vider le réservoir trop rapidement, et de faire en sorte de se retrouver dans une situation où il n'est plus possible de fournir la quantité d'eau requise pour permettre le fonctionnement des turbines, causant une baisse trop élevée de la production d'électricité. Au contraire, une politique plus restrictive peut entraîner une accumulation d'eau trop élevée et engendrer des débordements, ce qui n'est pas non plus souhaitable. On vient ainsi de décrire la principale difficulté du gestionnaire, et tout l'enjeu de trouver une politique optimale.

Le système hydroélectrique de Kemano est situé en Colombie britannique et alimente une centrale électrique permettant de fournir l'usine d'aluminium située à Kitimat en électricité. Cette dernière est principalement utilisée dans le procédé d'électrolyse de l'aluminium. La figure 2.2 montre un plan du système. Il est composé de deux éléments distincts : le réservoir de Nechako et la centrale électrique de Kemano. Le but ultime est d'assurer l'approvisionnement énergétique de la centrale de façon efficace tout en s'assurant de la sécurité du public. On retrouve aussi le lac de Cheslatta ainsi que la ville de Vanderhoof en aval qui sont à risque d'inondation.

Le réservoir situé en amont du barrage emmagasine l'eau qui sera ensuite envoyée vers les turbines et Skins Lake : c'est le réservoir de Néchako, qui emmagasine l'eau, et qui se vide à la vitesse choisie par l'opérateur, que nous appellerons administrateur dans le reste du texte. Le nom "réservoir" peut prêter à confusion car il ne s'agit pas d'une unique cuve qui emmagasine l'eau, mais plutôt d'un ensemble de lacs. Le point initial se situe au niveau du barrage de Kenney Dam, barrage qui permet justement à l'eau de s'accumuler, et se rend jusqu'au lac Tahtsa situé à environ 150 kilomètres à l'ouest. L'eau évacuée peut-être envoyée à deux endroits. Premièrement, vers la centrale de Kemano située à l'ouest du réservoir, à travers un long tunnel d'une quinzaine de kilomètres à côté du lac Tahtsa. Sinon, il est possible d'utiliser le déversoir de Skins Lake qui dirige l'eau vers la rivière Nechako, offrant ainsi un habitat propice à la faune et la flore dudit lac, ainsi qu'une possibilité d'évacuation du trop plein. Le déversoir est situé à mi-chemin entre le barrage de Kenney Dam et le lac de Tahtsa. Le surplus d'énergie est revendu à BC Hydro. Le but principal de l'administrateur est de permettre la production d'aluminium, tout en s'assurant de la sécurité du public. De plus, il doit prendre en compte le caractère stochastique des apports naturels, et trouver un moyen d'ajuster ses décisions en fonction de cela.

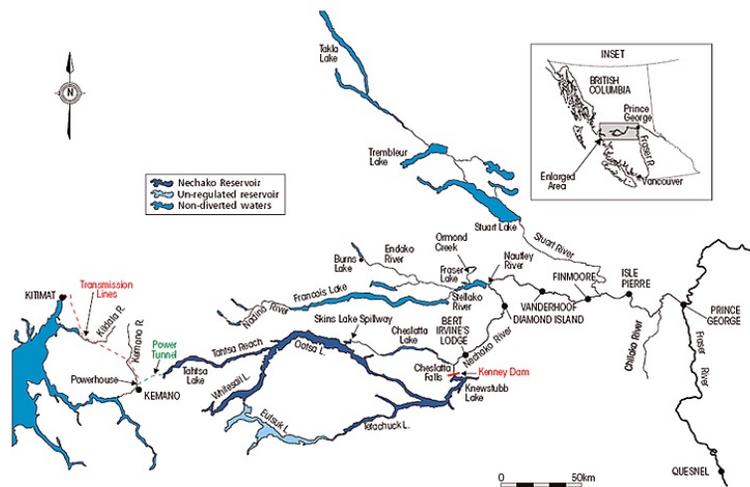


FIGURE 2.2 – Plan du système hydroélectrique de Kemano

2.2 Modélisation mathématique d'un système hydroélectrique

La section précédente proposait une description générale du fonctionnement du système. Dans cette partie, nous passons en revue les différentes limites et contraintes. Dans une autre sous-section, on ajoute une introduction aux réseaux de neurones qui permettront de résoudre notre problème d'optimisation.

2.2.1 Contraintes et optimisations du système de Kemano

Rappelons dans un premier temps que l'objectif de l'administrateur est d'assurer l'approvisionnement énergétique, tout en prenant en compte les différentes contraintes présentées dans cette section.

Des apports naturels $a_t^{(N)}$ tels que la pluie et la fonte des neiges remplissent le réservoir de Nechako tout au long de l'année, ce qui fait que la quantité d'eau qui lui parvient est très différente selon les saisons. C'est cette partie stochastique qui complique la prise de décision au fil du temps. En prenant en compte cette partie, ainsi que d'autres détails expliqués plus loin dans cette section, l'administrateur doit décider des différents débits relâchés du réservoir. Il devra donc faire un choix pour :

Le débit à envoyer vers les turbines $x_t^{(K)}$

Le débit à envoyer vers le déversoir de Skins Lake $x_t^{(S)}$

À noter : le terme 'apport' désigne les arrivées d'eau naturelles, donc des variables exogènes et aléatoires. Les 'débits' sont, quant à eux, les sorties d'eau choisies par l'administrateur et qui sont dirigées vers le déversoir et les turbines, donc des variables endogènes. De plus, chaque 'période', qui sera utilisée tout au long de ce mémoire, représente une durée de 3 jours. Une année est donc constituée de $T=122$ périodes, numérotées de 1 à T .

Plusieurs paramètres sont à prendre en compte avant d'établir un modèle mathématique :

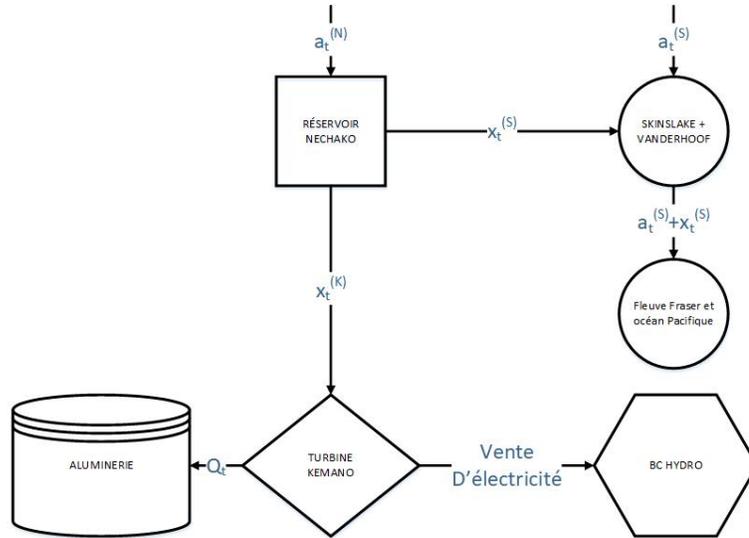


FIGURE 2.3 – Schéma du système hydroélectrique de Kemano

Premièrement, l'élévation N_t du réservoir. Cette variable a une influence sur l'électricité qui peut être produite. Plus l'élévation est grande, plus la vitesse de l'eau déversée dans les turbines sera élevée, de même que l'électricité produite. Le réservoir possède aussi un certain volume à chaque période V_t qu'il ne faut pas négliger car c'est lui qui permet de connaître la quantité d'eau qu'il est possible de relacher. En plus de cela, il faut comprendre que le réservoir de Nechako possède une limite supérieure $V^{(max)}$ qui ne peut pas être dépassée. Dans l'éventualité où le réservoir ne peut plus contenir l'eau, l'excédent sera envoyé vers Skins Lake. Le réservoir a aussi une limite inférieure $V^{(min)}$ qui ne peut pas être dépassée, sous peine d'une incapacité à produire l'électricité nécessaire à l'alimentation de l'usine de Kitimat. Il est en fait physiquement impossible de se retrouver en dessous de cette limite inférieure. Aussi, si le volume est trop faible, l'élévation minimale permettant à l'eau d'arriver suffisamment rapidement aux turbines et de produire l'électricité nécessaire est trop basse et force l'usine à arrêter ses activités, entraînant ainsi des coûts très élevés.

$$V^{(min)} \leq V_t \leq V^{(max)}$$

En ce qui à trait au débit vers Skins Lake, un minimum $d_t^{(Smin)}$ est requis pour

satisfaire les contraintes environnementales. On retrouve aussi un maximum $d_t^{(S_{max})}$. Ces valeurs varient au gré des saisons. Ces limites sont imposées par le gouvernement et ne sont donc transgressables sous aucunes conditions.

$$d_t^{(S_{min})} \leq x_t^{(S)} \leq d_t^{(S_{max})}$$

Les limites maximales et minimales vers Skins Lake selon les périodes se retrouvent dans la figure 2.4.

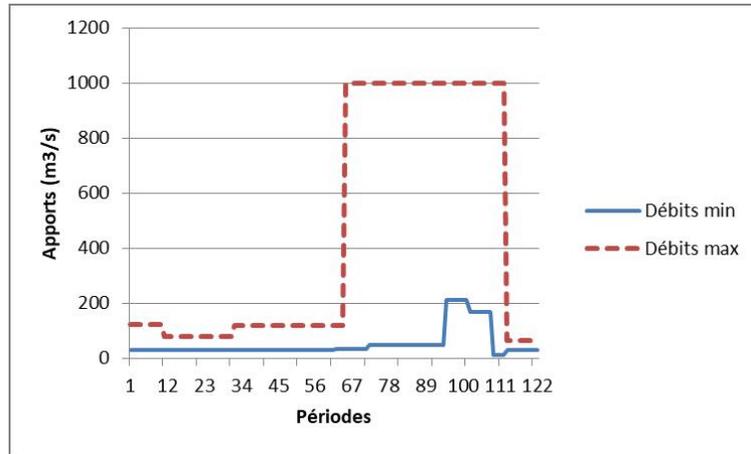


FIGURE 2.4 – Graphique des limites maximales et minimales selon les périodes

Pour le débit maximum en direction du lac, il faut ajouter les apports naturels d'eau qui se déversent vers Skins Lake $a_t^{(S)}$. La somme de ces apports avec les débits vers le lac choisis par l'administrateur ne devrait pas dépasser la limite de débordement q sous peine d'inondation de VanderHoof, la ville en aval.

$$a_t^{(S)} + x_t^{(S)} \leq q$$

Pour l'alimentation de l'aluminerie, la quantité d'électricité produite est une fonction notée Q_t , et dépend de deux éléments :

Le débit d'eau provenant du réservoir et se dirigeant vers les turbines $x_t^{(K)}$

L'élévation N_t .

Une matrice de conversion comprenant les deux variables précédentes permet l'interpolation des quantités d'électricité produites en Megawatts . Les interpolations

s'effectuent de deux façons : linéaire pour les fonctions à une variable et en utilisant des splines pour deux variables. Par ailleurs, pour obtenir le volume correspondant à une certaine élévation, il est nécessaire d'interpoler une fonction quasiment linéaire reliant les deux variables.

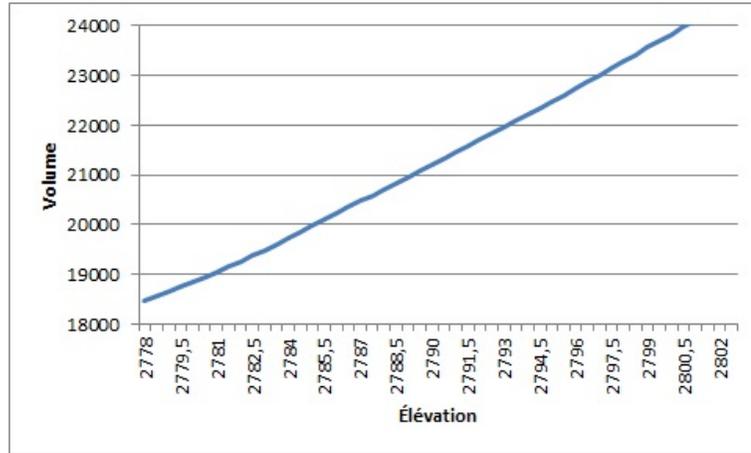


FIGURE 2.5 – Fonction de conversion des élévations en volume

L'administrateur a le contrôle (dans une certaine mesure) de la quantité d'eau qu'il peut libérer, mais l'élévation dépend de ses décisions précédentes ainsi que des apports naturels se dirigeant vers le réservoir. Pour que l'usine puisse fonctionner, elle doit recevoir une quantité d'électricité équivalente à 700 Megawatts à chaque période. Dans le cas contraire, la fermeture temporaire de l'aluminerie entraînera des pertes d'argent. Si nous réussissons à produire une quantité excédentaire, comme on ne peut la conserver pour des utilisations futures, elle sera vendue à BC Hydro. Une information importante que nous fournit la matrice de conversion est qu'au dessus d'un certain débit $x_t^{(K)}$, qu'on note $x_t^{(K_{min})}$, l'aluminerie sera suffisamment alimentée, et ce tant que l'élévation du réservoir est au moins supérieure à la limite minimale. De plus, à partir d'un certain débit noté $x_t^{(K_{max-mw})}$, toujours sous les mêmes conditions d'élévation, la quantité d'électricité générée n'augmentera plus. Enfin, le débit $x_t^{(K)}$ qu'il est possible d'envoyer n'est pas illimité. En effet, des contraintes physiques, telles que la taille des tuyaux par exemple, limitent la décision de l'administrateur.

Ce débit maximal est noté $x_t^{(K_{max})}$.

$$x_t^{(K_{min})} \leq x_t^K \leq x_t^{(K_{max})}$$

Ensuite, lors de la conversion des débits et de l'élévation en mégawatts, la matrice utilisée pour l'interpolation nécessite des débits en $\text{hm}^3/3\text{j}$, et non pas en m^3/s . Il faut donc les convertir en multipliant les débits par un facteur de conversion (Δ).

Maintenant que toutes les variables sont connues, on peut définir mathématiquement la fonction-objectif et sa composante. La fonction de production d'électricité est donnée par :

$$Q_t(x_t^{(N)}, N_t)$$

Pour rappel, $Q_t()$ est une fonction à deux variables dont les valeurs sont trouvées grâce à une matrice dont on fait une interpolation par spline. Cette fonction interpolée donne une surface quasiment plane. Elle est aussi convexe.

Enfin, la dernière contrainte est celle de la conservation de l'eau. En effet, le volume du réservoir est ajusté après chaque période selon la valeur des débits soutirés et celle des apports naturels qui le remplissent. Il faut donc tenir compte de ces changements à l'aide de la formule suivante :

$$V_{t+1} = V_t + a_{t+1}^{(N)} - \Delta * (x_t^{(K)} + x_t^{(S)})$$

Le but de l'administrateur est donc de maximiser :

$$E\left[\sum_{t=1}^T Q_t(x_t^{(N)}, N_t)\right]$$

Le tableau 2.1 résume les différentes variables du problème ainsi que leur unité.

noms	variables	unités
Apports naturels Néchako	$a_t^{(N)}$	m^3/s
Apports naturels Skins Lake	$a_t^{(S)}$	m^3/s
Débit vers la centrale	$x_t^{(K)}$	m^3/s
Débit vers Skins Lake	$x_t^{(S)}$	m^3/s
Débit min vers Skins Lake	$d_t^{(Smin)}$	m^3/s
Débit max vers Skins Lake	$d_t^{(Smax)}$	m^3/s
Débit limite avant inondation	q	m^3/s
Facteur de conversion	Δ	$m^3/s \rightarrow hm^3/3j$
Élévation du réservoir	N_t	pieds
Volume du réservoir	V_t	hm^3
Volume min du réservoir	V^{min}	hm^3
Volume max du réservoir	V^{max}	hm^3
Fonction de production	$Q_t()$	mégawatts

TABLE 2.1 – Tableau des différentes variables et de leur unité

Ci-dessous se trouve un résumé du problème et de ses contraintes :

$$E[\sum_{t=1}^T Q_t(x_t^{(N)}, N_t)]$$

s.c

$$V^{(min)} \leq V_t \leq V^{(max)}$$

$$a_t^{(S)} + x_t^{(S)} \leq q$$

$$d_t^{(Smin)} \leq x_t^{(S)} \leq d_t^{(Smin)}$$

$$x_t^{(Kmin)} \leq x_t^{(K)} \leq x_t^{(Kmax)}$$

$$V_{t+1} = V_t + a_{t+1}^{(N)} - \Delta * (x_t^{(K)} + x_t^{(S)})$$

Le problème étant décrit, nous comprenons que la cible est l'évaluation de l'espérance et sa maximisation. Le chapitre suivant expose les différentes méthodes utilisées pour répondre à cette question. Avant cela, il est indispensable de définir les réseaux de neurones qui seront au coeur de notre programme d'optimisation et d'évaluation.

Chapitre 3

Résolution du problème de gestion hydroélectrique

Ce nouveau chapitre décrit la façon dont nous allons résoudre notre problème mathématique. La première section explique le processus d'optimisation global, alors que la section qui suivra décrira notre utilisation des réseaux de neurones dans notre problème.

3.1 Algorithme de résolution et optimisation

Pour résoudre le problème du système Kemano, nous avons besoin d'un algorithme capable d'optimiser la fonction de revenus obtenus grâce à la production d'électricité. L'idée est de trouver une politique qui serait capable de résoudre le problème présenté à la section précédente. Pour ce faire, nous nous servons d'un ensemble de données simulées.

La matrice de la table 3.1 contient toutes les données utilisées en entrées dans notre algorithme. Chaque colonne représente un ensemble transmis à chaque période. Ainsi, chaque ensemble de trois variables contenu dans une colonne va passer dans le réseau de neurones pour que ce dernier puisse fournir à l'administrateur les débits

$$\begin{bmatrix} a_1^{(N)} & a_2^{(N)} & \dots & a_{122}^{(N)} \\ 1 & 2 & \dots & 122 \\ V_1 & V_2 & \dots & V_{122} \end{bmatrix}$$

TABLE 3.1 – Matrice des données utilisées pour l’optimisation de la politique

d’eau. Attention : la matrice 3.1 n’est pas connue au complet dès le début de la procédure. Elle est construite au fur et à mesure, car seul les volumes allant de la période 0 à i sont connus. Le volume $i + 1$ n’apparaîtra que lorsque l’ensemble des variables de la période i auront été utilisées dans le réseau de neurones, et que les débits auront été choisis.

La fonction de production d’électricité $Q_i(x_i^{(N)}, N_i)$ entre alors en jeu et renvoie la quantité de mégawatts produite. Pour cela, on utilise les débits $x_i^{(K)}$ choisis par la politique ainsi que l’élévation du réservoir, puis l’interpolation d’une table fournit les mégawatts qui en découlent. On obtient alors les bénéfices ou les pertes pour la période ciblée. On réitère cette opération pour chaque période (122) avec un nombre n de scénarios choisis au préalable.

La politique permettant à l’administrateur de choisir les débits est un réseau de neurones. La section suivante précise leur fonctionnement et leur utilisation.

3.2 Réseau de neurones en gestion hydroélectrique

La première partie de cette section se concentre sur une introduction aux réseaux de neurones, permettant ainsi de comprendre leur fonctionnement général. La deuxième partie décrira leur utilisation dans notre problème de gestion.

3.2.1 Introduction aux réseaux de neurones

Les réseaux de neurones artificiels (RNA) sont des outils qui s’inspirent des réseaux de neurones biologiques et de leurs liaisons synaptiques. Ils permettent de faire des approximations de fonctions non linéaires. Un exemple de leur utilisation

est la tentative d'identifications de tumeurs sur les radios aux fins de traitement. L'objectif est de permettre au réseau de trouver les limites exactes des tumeurs et des organes sur les scanners, facilitant le traitement. La figure 3.1 montre un réseau de neurones simple non récurrent. Cela signifie qu'une sortie ne peut pas influencer une entrée. Ce réseau est composé d'une couche d'entrée avec 3 unités d'entrées, une couche de sortie avec 2 unités de sorties ainsi qu'une couche cachée comprenant 3 neurones. Ce que nous appelons neurones sont en fait des entités qui utilisent les entrées qu'on leur fournit pour en calculer une somme pondérée qui sera retransmise à ce qu'on appelle une fonction d'activation. Cette dernière transforme le résultat et permet d'obtenir une sortie qui sera utilisée dans l'unité suivante, réutilisée pour l'apprentissage ou simplement retournée à l'utilisateur.

La couche d'entrée agit sur toutes les données initiales (appelées aussi signaux d'entrées) qui sont transmises au réseau. Elles sont notées X_0, X_1, \dots, X_n , n étant le numéro de l'entrée correspondante. La deuxième partie contient les couches cachées. Elle sont composées, dans le cas le plus simple, d'un seul neurone, mais il arrive de n'avoir qu'une seule couche composée d'un ou de plusieurs neurones. Cela dépend de l'objectif du réseau de neurones ainsi que du nombre d'entrées et de sorties. Dans notre cas, il est composé de plusieurs neurones qui permettent la transformation des données d'entrées. Enfin, la dernière couche contient les sorties qui permettront, soit d'entraîner le réseau, ou simplement d'obtenir les résultats de la transformation qui est faite à l'intérieur du RNA.

Chacun des neurones est relié à l'autre par des poids notés $\omega_{n,m}^{(j)}$, $i = 1$ correspondant à un poids allant de l'entrée vers la couche cachée, $j = 2$ de la couche cachée à celle de sortie, n le numéro de l'entrée (sortie) et m celui du neurone correspondant (dans la figure 3.1, pour la couche cachée, $m=1,2,3$ car on y retrouve 3 neurones). Ces poids permettent de pondérer les données reçues et d'identifier leur importance par rapport aux autres. Ils sont habituellement déterminés par le réseau lors de l'apprentissage. À l'aide des données fournies en entrée et des résultats obtenus lors de l'entraînement, les poids vont être modifiés jusqu'à permettre aux

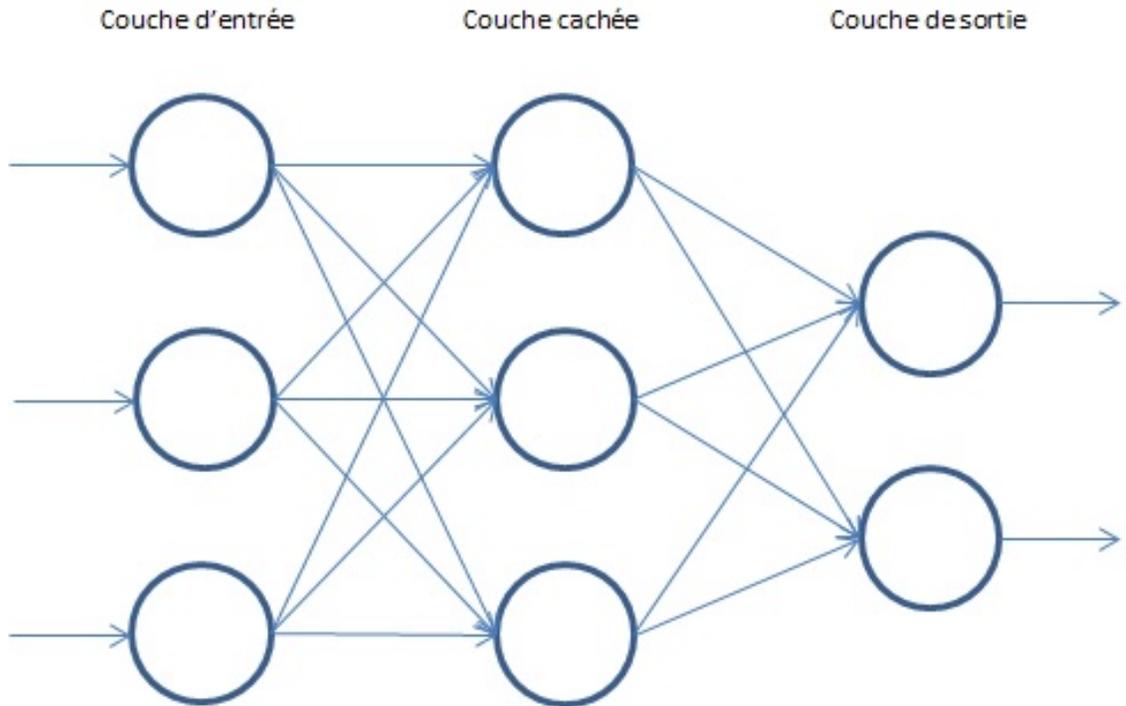


FIGURE 3.1 – Réseau de neurone avec 3 unités d’entrées, 2 unités de sorties et une couche cachée de 3 neurones

résultats de se rapprocher au maximum des valeurs de sorties recherchées. Mais ce n’est pas le seul paramètre qui agit sur les résultats. Il faut aussi transformer les sommes pondérées à l’aide de ce qu’on appelle une fonction d’activation notée $g_m(x)$, avec m représentant le $m^{ième}$ neurone de la couche d’arrivée. Par exemple, si l’on passe de la couche d’entrée à la couche cachée, alors m prendra les valeurs 1,2 ou 3 selon le neurone atteint par le poids partant de la couche d’entrée. Elle est le plus souvent non-linéaire et permet d’agir sur les signaux pour fournir une sortie. Il existe plusieurs types de fonction qui ont des utilités différentes selon les situations. Le plus important est de connaître nos données, savoir comment nous voulons entraîner notre réseau et quels types de sorties sont désirées. Par exemple, si l’on veut utiliser des optimisations basées sur le gradient, il est important que la fonction d’activation soit partout différentiable (comme la fonction sigmoïde ou la tangente hyperbolique, par exemple.) Une fonction d’activation générale est décrite

par l'équation suivante :

$$g_m(\omega_0 + \sum_{j=1}^n \omega_{j,m,i} X_j)$$

avec $\omega_{j,m}$ le poids reliant le neurone n de la couche de départ au neurone m de la couche d'arrivée et X_n l'entrée appliquée au neurone n . Il est possible d'utiliser des fonctions d'activations différentes pour chaque couches. Enfin, ω_0 représente le biais. Ce dernier est simplement une nouvelle entrée toujours égale à 1. Le biais permet de mieux comprendre l'effet des autres entrées puisque ces dernières changent constamment, et l'effet de la modification des poids lors de l'instance la plus récente peut-être compensé par de nouvelles entrées. Pour bien comprendre comment notre propre réseau fonctionne, il faut faire une distinction entre 3 types d'apprentissages possibles : supervisé, non-supervisé et par renforcement.

Apprentissage supervisé

Dans le premier cas, les entrées et les sorties sont connues à l'avance. Le but du réseau va être de devenir capable d'identifier les sorties correspondantes aux entrées fournies. Pour ce faire, à chaque passage dans le réseau, ce dernier va enregistrer l'erreur entre le résultat qu'il aura lui même trouvé et celui attendu, et ajuster les poids afin de réduire cette erreur lors du prochain passage, et ceci pour chaque ensemble entrées-sorties. On utilise le plus souvent une méthode des moindres carrés. Le but est d'être capable, en donnant de nouvelles entrées au réseau, d'obtenir de bons résultats. Par exemple, dans l'utilisation de la reconnaissance de l'écriture manuscrite ou encore d'images. On fournit un ensemble de chiffres écrits à la main ainsi que la valeur attendue pour chacun d'eux. Le réseau s'entraîne alors sur ces images et il devra être capable de retrouver, sur de nouveaux ensembles de chiffres, la valeur correspondante.

Apprentissage non-supervisé

Dans le cas de l'apprentissage non-supervisé, on ne connaît pas les sorties au préalable. Le but est de classer les différentes entrées dans des groupes sans connaître à l'avance la répartition, le nombre de groupes et leurs distinctions. Il doit être capable de regrouper les données sans connaissances préalables. Le réseau va regrouper les données dans des groupes qu'il estime similaires. Par exemple, on peut chercher des anomalies dans des transmissions en regardant celles qui semblent différentes. Le réseau doit réussir à créer (au moins) deux groupes : l'un avec les transmissions sans anomalies, et l'autre avec des transmissions anormales.

Apprentissage par renforcement

Enfin, dans le dernier cas, le réseau va ajuster les poids en espérant obtenir de meilleurs résultats à chaque itération, et va retenir ceux qui optimisent la décision. C'est d'ailleurs cette méthode que nous utilisons dans ce mémoire. Le chapitre suivant explicite les méthodes employées pour résoudre notre problème.

3.2.2 Utilisation des réseaux de neurones comme politique de gestion

Nous faisons appel à un réseau de neurones afin de trouver les poids qui maximisent les bénéfices obtenus de la production et de la vente de l'électricité produite grâce au passage de l'eau dans les turbines. On ajuste aussi ces poids afin de trouver une politique qui résout notre problème de gestion. Notre réseau de neurones est la politique qui va fournir à l'administrateur les débits à envoyer vers Skins Lake et Kemano. Le réseau de neurones est constitué de trois couches aux usages différenciés. Une première couche constituée de 3 neurones correspondants aux 3 entrées à chaque période : le volume du réservoir V_i , la période i et les apports naturels en direction du réservoir $a_i^{(N)}$. On justifie l'utilisation de ces 3 entrées spécifiques pour les raisons suivantes :

V_i : Le volume est la variable qui sera la plus influencée par les débits et les apports. En effet, l'ajustement à chaque période nous donne une nouvelle valeur du volume à prendre en considération. De plus, les contraintes sur ses limites décrites dans le chapitre précédent doivent être respectées en tout temps, et il est nécessaire d'adapter les décisions tout au long du processus pour éviter de mettre en péril l'approvisionnement énergétique futur.

i : On sait qu'il existe un caractère saisonnier en ce qui à trait aux apports naturels. Il faut donc prendre les périodes en compte puisque selon le moment de l'année, les probabilités des grandeurs des apports seront différentes. Par exemple, vers la fin de l'hiver, la fonte des neiges aura tendance à grandement augmenter la valeur des apports d'eau naturels.

$a_i^{(N)}$: Étant la partie stochastique de ce problème, et intervenant dans le volume du réservoir, il est obligatoire d'inclure cette variable.

Une deuxième couche, appelée couche cachée, est constituée de plusieurs neurones. Nous ne sommes pas plus précis concernant ce nombre puisqu'il sera modifié au fil des tests pour trouver le meilleur équilibre précision/temps de calcul. À cette étape, nous nous contenterons de préciser qu'il se situe entre 5 et 30. Enfin, la dernière couche, appelée couche de sortie, contient quatre neurones qui vont correspondre à chacune des décisions prédéfinies. À ce stade, il n'est pas nécessaire de détailler les politiques car elles seront décrites dans le dernier chapitre. Dans la deuxième couche, nous utilisons la tangente hyperbolique comme fonction d'activation. En revanche, une fonction d'activation softmax (ou fonction exponentielle normalisée) sera privilégiée pour la couche de sortie. Cette fonction permet, de part sa forme, d'obtenir un nouvel ensemble de réels positifs dont la somme est l'unité, en fournissant un vecteur $z = (z_1, z_2, \dots, z_k)$ de K nombres réels.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$

Son intérêt est de permettre à notre réseau de nous retourner exactement le nombre

de sorties correspondant à nos différentes actions possibles et de pouvoir ainsi choisir celle dont la valeur retournée est la plus élevée. Nous obtenons donc une probabilité de poser une certaine action. Dans notre problème, le vecteur z de la fonction softmax représente l'ensemble des entrées fournies à notre réseau de neurones, et le vecteur $\sigma(z)$ celui des probabilités de poser une des actions proposées. Lorsque le réseau reçoit un ensemble de données, ces dernières vont passer dans chacune des couches et être transformées à l'aide des poids. Ceux-ci sont, pour la toute première instance, tous égaux. Après être passés dans toutes les périodes de tous les scénarios choisis, ils seront modifiés et une nouvelle instance sera relancée.

Par ailleurs, une fois que l'administrateur reçoit les probabilités d'actions fournies par le réseau, avant d'appliquer directement les débits correspondant, il doit parfois réaliser un dernier ajustement par rapport aux débits à envoyer. En effet, il est par exemple possible que le volume du réservoir dépasse le maximum permis. Dans ce cas, le surplus est envoyé, par défaut, vers le déversoir de Skins Lake. Concrètement, cela aura un effet sur la valeur de $x_i^{(S)}$ qui sera plus élevée que celle fournie par le réseau.

Il est important de comprendre que les poids de notre réseau ne sont pas optimisés par ce dernier, mais plutôt par un autre algorithme. En effet, il faut transmettre les bénéfices à un algorithme capable de les améliorer. Nous avons opté pour celui de Nelder-Mead, qui va recevoir comme variables les poids du réseau, et les modifier selon son processus afin d'améliorer les bénéfices.

L'utilisation de Nelder-Mead se justifie par le fait qu'il permet d'optimiser une majorité de fonctions compliquées puisqu'il n'exige pas de dérivée. Nelder-Mead va donc prendre les poids de l'instance actuelle ω_k et retourner de nouveaux poids ω_{k+1} jusqu'à converger et offrir ω^* , qui seront les poids optimaux. Ici, k représente le nombre de fois que Nelder-Mead modifie les poids. Il est important de ne pas confondre k avec i , qui représente une période. Plus précisément, notre réseau de neurones va recevoir les entrées à chaque période (122 au total) pour un certain nombre de scénarios choisis au préalable. Lors de chacune de ces périodes, après

l'obtention des probabilités d'actions, l'administrateur va choisir les débits correspondants, et ajuster le volume. Il en résulte alors un bénéfice en dollars provenant de la génération d'électricité. Une fois que toutes les périodes ont été traversées, la somme des bénéfices obtenus est envoyée à Nelder-Mead, ainsi que les poids actuels du réseau de neurones. Nelder-Mead ajuste alors les poids selon son processus pour tenter d'obtenir, à l'instance suivante, un meilleur bénéfice.

Mathématiquement, Nelder-Mead obtient les poids ω_k du réseau de neurones et maximise alors la fonction $f(\omega_{k+1})$. Bien que nous ne connaissions pas la fonction f , sa valeur correspond à la somme des bénéfices obtenus à la fin du passage dans toutes les périodes. Nelder-Mead peut alors modifier les poids et comparer les bénéfices obtenus grâce à $f(\omega_{k+1})$. L'optimisation se fait donc sur l'espérance des bénéfices :

$$E_i[R_i]$$

Pour synthétiser ce qui précède, dans un premier temps, nous fournissons des poids ω_k et les entrées V_i, a_i et i à chaque période au réseau de neurones. Ce dernier les transforme en un vecteur dont la somme des valeurs est 1, et la plus haute valeur de ce dernier nous fournit les débits à utiliser. Les débits sont ajustés pour correspondre aux critères cités dans le chapitre précédant et une certaine quantité d'électricité Q_i est obtenue, elle-même convertie en bénéfices en dollars R_i . Ces étapes sont répétées avec les mêmes poids jusqu'à ce que l'ensemble des périodes de tous les scénarios choisis soit traversé. Nous obtenons alors une somme de bénéfices pour toutes les périodes. Ces mêmes bénéfices sont alors transmis à un algorithme de Nelder-Mead qui va modifier les poids. Après cette étape nous recommençons depuis le début avec des données identiques, excepté pour les poids ω_{k+1} qui viennent d'être modifiés. Les instances se succèdent jusqu'à ce que Nelder-Mead converge vers un optimum. La figure 3.2 résume le processus. Le chapitre suivant décrira la mise en oeuvre de cette procédure et développera différents résultats numériques complétés de leur analyse.

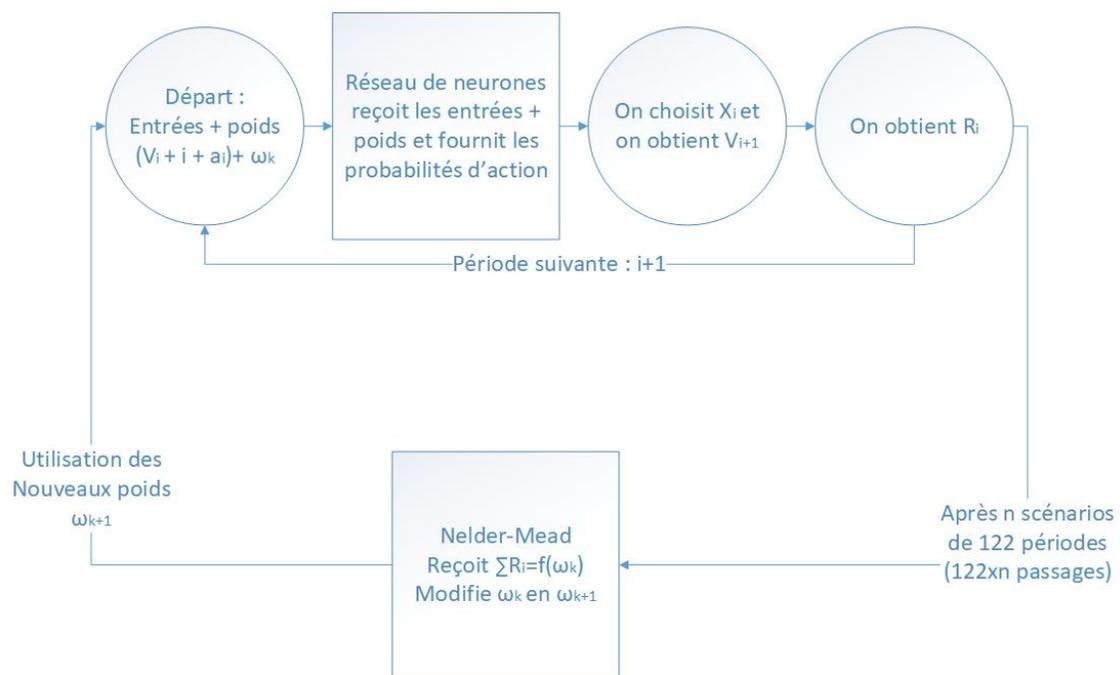


FIGURE 3.2 – Schéma de l’algorithme complet d’optimisation

Chapitre 4

Mise en oeuvre et résultats numériques

Cet ultime chapitre présente les différentes procédures mises en oeuvre pour obtenir les mégawatts issus de notre algorithme. Dans la première section, nous expliquons la manière dont le code a été construit autour du RNA et la façon dont les paramètres et les débits sont modifiés pour s'ajuster au problème et aux différentes limites imposées par le système. Ensuite, nous présentons un ensemble d'actions possibles et de méthodes d'optimisation qui ont été testées tout au long du processus de recherche. L'objectif de ces tests était de déterminer lesquelles étaient les plus performantes et, donc, offraient les meilleurs résultats. Pour ce faire, nous comparerons les différents bénéfices et mégawatts obtenus selon chacune d'elles, ainsi que la vérification du respect de certains critères décrit plus loin dans ce chapitre. Enfin, les dernières sections présenteront des tests de sensibilité en lien avec le réseau de neurones, ainsi que des tests supplémentaires.

4.1 Contraintes de fonctionnement et mégawatts

Pour optimiser les mégawatts, nous allons nous baser sur les bénéfices obtenus lors de la vente des mégawatts excédentaires. En effet, les mégawatts et les bénéfices étant étroitement liés, il est plus simple de se concentrer sur ces derniers. De plus, comme seul les mégawatts excédentaires (c'est-à-dire en plus de ceux permettant l'alimentation de l'usine) augmenteront les bénéfices de façon proportionnelle, il sera simple de vérifier si les mégawatts produits sont suffisants. Pour permettre à notre algorithme d'optimiser correctement les poids du réseau de neurones, il est important d'ajuster les gains provenant de la génération de l'électricité et des pertes conséquentes au non respect des limites du système. Par gains, nous entendons les bénéfices provenant de la vente d'électricité lorsque l'usine est suffisamment alimentée. Pour les pertes, nous parlons ici de la réduction des bénéfices lorsque certains critères ne sont pas respectés, comme par exemple la limite du volume du réservoir, les débordements ou encore les inondations. Nous avons déjà abordé la valeur des mégawatts produits lors du passage des débits d'eau dans les turbines, et nous proposons, maintenant, la formule permettant d'obtenir les bénéfices dépendants des mégawatts vendus à BC Hydro :

$$R_i = \mathbb{1}[Q_i() \geq 700] * (Q_i() - 700) * W$$

La fonction identité située à gauche donne 1 uniquement dans le cas où la production d'électricité $Q_i()$ atteint au moins les 700 mégawatts. Si c'est le cas, les bénéfices seront proportionnels au nombre de mégawatts supérieurs à 700, ce qui est donné par la deuxième partie de l'équation $(Q_i() - 700)$. Dans la constante W , on retrouve d'abord le gain horaire de 180 dollars. Comme les périodes correspondent à trois jours, on multiplie ce gain horaire par 72 heures pour obtenir $W = 180 * 72 = 12960\$$. L'usine réclame une quantité précise d'électricité pour fonctionner : 700 Mégawatts. Donc seuls ceux produits au dessus de 700 seront revendus. Ainsi, chaque mégawatt produit au dessus des 700 requis offre un bénéfice supplémentaire de 12960\$. Point important, dans le cas où une valeur inférieure à 700 mégawatts serait produite,

les bénéfiques deviendraient nuls, et non pas négatifs, d'où la présence de la fonction identité. Si jamais le réservoir se retrouve à sa limite inférieure, alors les débits ne sont plus suffisants pour alimenter l'usine, et les bénéfiques sont nuls, jusqu'à retrouver un volume permettant d'avoir des débits qui fournissent au moins les 700 mégawatts requis.

En lien avec cette dernière contrainte, nous voulons limiter les années où le réservoir se trouve à sa limite inférieure. Nous visons donc un maximum de 5 années sur 60 où ce cas se produit. Il est aussi possible que la ville de VanderHoof commence à s'inonder si la somme des débits provenant du réservoir et des apports naturels vers SkinsLake, lors d'une période, dépasse les 550 m³/s. Jusqu'à 600 m³/s, nous n'imposons pas de pénalités. Mais dès lors que l'on dépasse cette dernière limite, une pénalité proportionnelle à chaque mètre cube d'eau excédentaire est imposée. Dans ce cas, les bénéfices de la production d'électricité seront réduits. Cette pénalité est expliquée juste après la formule de retrait aux bénéfices proportionnelle à chaque m³/s qui dépasse la limite des 600 :

$$Pertes = \mathbb{1}(V_{i+1} - V^{max} + x_i^{(S)} > 600\text{m}^3/\text{s}) * (V_{i+1} - V^{max} + x_i^{(S)} - 600) * L$$

Dans cette équation, la fonction identité à gauche ne s'active que si la somme des débits vers SkinsLake et des débordements du réservoir dépasse les 600 m³/s. Les pertes proportionnelles à chaque m³/s excédentaire sont indiquées dans la deuxième partie de l'équation. De manière similaire à W , L représente une perte horaire de 360 dollars multipliée par 3 jours pour s'ajuster aux périodes. Nous avons alors $L = 360 * 3$. En lien avec cette dernière contrainte, il est utile de préciser qu'on considère qu'une année (scénario) est en inondation à partir du moment où au moins une de ses périodes, sur les 122, se retrouve avec un débit supérieur à 600 m³/s. Par expérience, nous savons qu'il est impossible de n'avoir aucune inondation. On retrouve toujours au moins 2 années dans des situations critiques forçant l'inondation. Aussi visons nous un taux d'inondation ne dépassant pas les 10%, c'est à dire que sur un ensemble de 60 années, nous désirons connaître un maximum de 6 années contenant une ou

plusieurs périodes répondant au critère d'inondation. Nous ajoutons par ailleurs un autre facteur de pénalité de 315. Ce dernier se base sur des essais répétés avec différentes valeurs de pénalité. Des valeurs plus faibles entraînaient des inondations trop nombreuses, et le contraire vidait le réservoir sans laisser suffisamment d'eau pour les périodes suivantes. La fonction de pertes devient alors :

$$Pertes = \mathbb{1}(V_{i+1} - V^{max} + x_i^{(S)} > 600\text{m}^3/\text{s}) * (V_{i+1} - V^{max} + x_i^{(S)} - 600) * L * 315$$

Il est aussi possible que le volume du réservoir dépasse la limite maximale. Dans ce cas, il n'y a pas de pénalité mais l'excédent d'eau est directement envoyé vers SkinsLake. Deux problèmes apparaissent alors :

-Nous n'avons plus le contrôle sur les débits pendant cette période (et possible-ment quelques suivantes selon la valeur des apports naturels.)

-Il est possible de se retrouver en situation d'inondation.

Une autre contrainte est la limite inférieure du réservoir. Il est indispensable que le volume du reservoir ne tombe pas en dessous de 19979.52 hm^3 . Dans le cas contraire, les turbines aspirent de l'air et cela empêche la génération d'électricité. Enfin, il existe un dernier point à aborder. Les 3 entrées que nous envoyons au réseau de neurones ont des échelles très différentes. Les volumes se trouvent au niveau des 20000 hm^3 , les périodes entre 0 et 1, et les apports entre 50 et $1200 \text{ m}^3/\text{s}$. Comme dans un réseau le résultat est dépendant des poids, mais aussi de la valeur des données d'entrées, les volumes prennent une place bien plus importante que les autres. Nelder-Mead n'arrivera donc pas à trouver un bon optimum car seuls les volumes auront une influence. Il faut donc uniformiser nos données pour éviter que seul les volumes aient une influence sur les sorties du RNA. L'uniformisation est simplement réalisée en ramenant les valeurs entre 0 et 1, avec 0 la plus petite et 1 la plus grande.

4.2 Décisions et méthodes d’optimisations

Pour alimenter notre code, nous possédons, pour les apports naturels, 5 ensembles d’apports d’eau qui sont des séries synthétiques fournies par la compagnie (nommés apports simulés 1 à 5), ainsi qu’un ensemble d’apports historiques entre les années 1957 et 2016. Par ensemble, nous entendons des apports d’eau sur 60 années, avec 122 périodes par année, pour un grand total de 7320 apports par ensemble. Nos optimisations se feront en choisissant un échantillon d’un des ensembles, puis nous testerons les résultats hors-échantillons, c’est à dire sur les autres ensembles disponibles. Nous avons donc choisi le premier ensemble simulé pour l’optimisation des poids, et nous vérifierons la qualité sur tout les autres autres ensembles. À partir de maintenant, sauf si mentionné, les scénarios utilisés pour déterminer les actions sont ceux du premier ensemble.

4.2.1 Choix des débits vers Skins Lake et Kemano

L’optimisation se fait sur plusieurs périodes, et pour chacune d’elles, l’administrateur doit prendre une décision pour les débits qu’il déploie vers les turbines et vers SkinsLake. Ce sont là les seules variables sur lesquelles il a un contrôle, et doit donc adapter sa décision par rapport aux autres informations du problème (niveau, apports, contraintes.) Son but est d’obtenir le bénéfice maximal. Celui-ci augmente lorsque la production dépasse les 700 MégaWatts, et chute drastiquement lorsqu’on se retrouve en deçà de cette valeur.

Pour réussir, nous discrétisons les décisions possibles de l’administrateur. Il dispose d’un choix de quatre actions qu’il peut utiliser à chaque période. Dans chacune d’elle, on propose un certain débit vers SkinsLake et vers les turbines. Elles sont décrites ci-dessous, en commençant par la plus restrictive.

Pour la première, le débit vers SkinsLake $x_t^{(S)}$ est le minimum légal pour la période en question, soit $d_t^{(Smin)}$, et le débit vers les turbines $x_t^{(N)}$ est le minimum permettant de faire fonctionner l’usine : $x_t^{(Nmin)}$ ou $110 \text{ m}^3/\text{s}$.

La deuxième action permet toujours d’avoir un débit pour SkinsLake équivalent au minimum légal pour la période, mais envoie un débit permettant d’obtenir le maximum d’électricité possible vers les turbines tant que l’élévation est au dessus de 2785 pieds, soit 148 m³/s.

La troisième action permet de vider plus rapidement le réservoir en proposant, vers SkinsLake, le débit $d_t^{(S_{max})}$, soit le maximum légal pour la période, dans la limite de 550 m³/s, et le débit minimum vers les turbines permettant d’avoir toujours la production maximale de mégawatts selon l’élévation, c’est-à-dire un débit de 148 m³/s.

Enfin, la dernière action offre la possibilité de vider très rapidement le réservoir en permettant des débits maximums des deux côtés : le maximum légal $d_t^{(S_{max})}$ pour SkinsLake (sans dépasser 550 m³/s), et 170 m³/s vers les turbines correspondant à la capacité maximale des tuyaux.

Le tableau 4.1 résume les différentes actions possibles.

Actions	Débits turbines	Débits Skins Lake
Action 1	110 m ³ /s	minimum légal pour la période
Action 2	148 m ³ /s	minimum légal pour la période
Action 3	148 m ³ /s	maximum légal pour la période
Action 4	170 m ³ /s	maximum légal pour la période

TABLE 4.1 – Décisions possibles pour le système hydroélectrique de Kemano

Pour rappel, dans les actions 3 et 4, les débits vers les turbines, bien que différents, produiront la même quantité d’électricité. La seule raison d’adopter la 4^{ieme} action est dans l’éventualité d’une inondation importante.

À chaque période, l’administrateur choisit une action proposé par le réseau de neurones, et selon l’énergie générée, obtient un bénéfice en dollars, auquel on retire les possibles pénalités en cas d’inondations. On ajuste alors le volume du réservoir en retirant les débits et en ajoutant les apports naturels de la période suivante. On répète alors l’opération pour toutes les autres périodes. L’ajustement du volume se fait à l’aide de la formule suivante :

$$V_{i+1} = a_{i+1}^{(N)} + V_i - (x_i^{(K)} + x_i^{(S)}) * \Delta$$

Attention, V_{t+1} n'est pas forcément le volume final à la fin de la période en cours. En effet, par la suite, des ajustements sont effectués afin d'éviter le débordement du réservoir ou la chute de son volume sous la limite inférieure. Le volume s'exprime alors de la façon suivante :

$$V_{i+1} = \min(a_{i+1}^{(N)} + V_i - (x_i^{(K)} + x_i^{(S)}) * \Delta, V_{max})$$

Pour ce qui est des débits, nous commençons par poser :

$$\beta = V_{max} - (a_{i+1}^{(N)} + V_i - (x_i^{(K)} + x_i^{(S)}) * \Delta)$$

Si β est positif, cela signifie que le réservoir dépasse sa limite supérieure, et qu'il faut ajouter cette valeur de débordement aux débits de SkinsLake. Ils deviennent alors :

$$x_i^{(S*)} = \max(x_t^{(S)} + \beta, x_t^{(S)})$$

Afin d'optimiser les poids du réseau de neurones, nous avons mis au point différentes méthodes. Dans chacune d'elles, nous choisissons plusieurs scénarios, et nous maximisons les bénéfices sur ceux-là. Une fois terminé, nous utilisons les poids trouvés pour tester sur un grand nombre de scénarios hors-échantillons et, enfin, pour vérifier la qualité des résultats (bénéfices et critère d'inondation). Le nombre de neurones utilisés dans notre réseau est initialement de 5. Ci-dessous ce retrouve le pseudo-code de tout le processus d'optimisation. Les méthodes et les résultats, quant à eux, sont résumés dans les sous-sections suivantes.

Pseudo-code du processus d'optimisation :

```

1 for iteration Nelder – Mead  $k \leftarrow$  premiere iteration to
  Derniere iteration do
2   for scenario  $\leftarrow$  scenario initial to scenario final do
3     for  $i \leftarrow 1$  to 122 do
4       Réseau de neurones reçoit les entrées et les poids;
5        $a_i^{(N)}$ ;  $V_i$ ;  $i$ ;  $\omega_k$ ;
6       Choix d'une des quatres actions et des débits;
7        $x_i^{(K)}$   $x_i^{(S)}$ ;
8       Obtention du nouveau volume;
9        $V_{i+1} = V_i + a_{i+1}^{(N)} - \Delta * (x_i^{(K)} + x_i^{(S)})$ ;
10      Ajustement du volume selon les contraintes du système;
11      if  $V_{i+1} > V^{max}$  then
12        |  $x_i^{(S)} = x_i^{(S)} * \Delta + (V_{i+1} - V^{max})/\Delta$ ;
13      end
14      if  $V_{i+1} < V^{min}$  then
15        | Ajustement des débits selon les contraintes vers Skins Lake;
16      end
17      Production MW  $\leftarrow Q_i(V_i, N_i, x_i^{(K)})$ ;
18      Benefices =  $700 - Q_i()$ ;
19    end
20  end
21  Les bénéfices et les poids sont retransmis à Nelder-Mead;
22  Modification des poids par Nelder-Mead;
23  if Nelder-Mead converge ou Nombre maximal d'itération then
24    | fin de l'algorithme;
25  else
26    | Nouvelle itération de Nelder-Mead  $k+1$ 
27  end
28 end
29 end

```

4.3 Méthodes d'optimisation

Nous présentons dans cette section une description de toutes les méthodes utilisées, ainsi que les résultats correspondants.

4.3.1 Méthode 1

La première méthode testée est la plus simple : sur les 60 scénarios, 3 sont sélectionnés, aléatoirement, ainsi qu'un volume de départ de 22000 hm³ (on s'arrête sur ce choix car il est entre les deux limites imposées pour les volumes du réservoir.) Avec les poids de départ, le réseau va passer dans les 3 années (122*3=366 périodes) et emmagasiner tous les bénéfices enregistrés pour les additionner à la fin de toutes les périodes. À chaque nouvelle instance, le volume de départ sera identique, soit 22000 hm³. Ce résultat est ensuite transmis à Nelder-Mead qui va ajuster les poids et relancer une instance jusqu'à converger. Ces tests ont aussi été réalisés pour des groupes de 5 et 7 scénarios. Pour tout le reste du texte, une instance est considérée comme complétée après avoir effectué 122 passages (car 122 périodes) dans le réseau de neurones multipliés par le nombre de scénarios choisis. Plus précisément, si nous choisissons n scénarios, une instance est complétée après $122 * n$ passages dans le réseau de neurones. À la fin d'une instance, on obtient une somme de bénéfices résultant de la production d'électricité, et Nelder-Mead utilise cette somme pour modifier les poids. Nous commençons avec des poids tous égaux à 1. Pour la première instance, l'algorithme passe dans les 366 périodes. Une fois les bénéfices obtenus, Nelder-Mead modifie les poids et repasse dans les 366 périodes, où les actions proposées vont alors changer selon la valeur des poids et des entrées.

Pour comparer, nous avons tenté l'expérience avec plusieurs groupes (10) de 3, 5 et 7 scénarios différents. Il faut préciser que lors de chaque optimisation, le passage ne s'effectue que dans un seul groupe.

Résultats méthode 1

Au vue du nombre limité de scénarios, un des risques de la méthode est le surapprentissage. C'est ce qui semble se produire puisque dans la majorité des cas, les poids retournés ne proposaient que l'action numéro 2 (à chaque période), sans aucune distinction. Dans les autres cas comportant un mélange de choix d'action selon les périodes, les tests hors échantillons ne produisaient pas de bons résultats, car trop spécifiques pour les petits groupes de scénarios. De plus, le volume de départ étant fixé lors de l'optimisation, un changement de ce dernier lors des tests provoquait des situations où les bénéfices tombaient bien en dessous des valeurs obtenues dans les essais précédents. Nous avons alors tenté d'ajouter des scénarios dans les méthodes suivantes.

4.3.2 Méthode 2

La deuxième méthode complique un peu le processus : Nous choisissons 10 groupes de 5 scénarios car 3 est un nombre trop faible et entraînerait un surapprentissage. L'algorithme passe d'abord dans le premier groupe (comme dans la première méthode) et enregistre les bénéfices. Ensuite, il passe dans le deuxième groupe pour faire la même chose, en conservant toujours un volume initial de 22000 hm³. Ce volume est choisi car il est situé à mi-chemin entre les volumes maximal et minimal autorisés. À la fin des 10 groupes, la moyenne de la totalité des bénéfices est renvoyée à Nelder-Mead qui modifie, alors, les poids pour tenter d'améliorer le résultat lors de la prochaine instance. Il s'agit d'entraîner l'algorithme sur beaucoup plus de scénarios, ici 50, afin qu'il puisse s'adapter à des années très différentes sur le plan des apports naturels.

Résultats méthode 2

Là encore, le principal problème provient du fait qu'à chaque itération de Nelder-Mead, aucune donnée ne varie. Comme le volume de départ et les scénarios sont

identiques, il y a, là encore, phénomène de surapprentissage, et donc une faible production de mégawatts lors des tests hors échantillons.

4.3.3 Méthode 3

La troisième méthode reprend à nouveau un seul groupe de 5 scénarios, mais utilise un volume de départ aléatoire compris entre 60% et 90% du volume maximal permis. Compte tenu du nombre limité de scénarios utilisés, le niveau de départ doit avoir une importance non négligeable, et le maintenir à 22000 hm³/s risque de causer à nouveau des problèmes de surapprentissage, voire même une difficulté au niveau de l'adaptation à d'autres volumes.

Avant de lancer une instance, un autre paramètre est modifié, à savoir, les poids initiaux. À partir de cette étape, et pour toutes les suivantes, nous effectuerons toujours deux tests : le premier, avec des poids tous égaux à 1 et le second, avec les derniers poids trouvés à la fin du premier test de la méthode en cours. Il s'agit d'une procédure de vérification de l'influence potentielle des poids initiaux.

Résultats méthode 3

Dans le cas où les poids sont de 1, la production de mégawatts est équivalente à ce qui est actuellement obtenu mais nous avons beaucoup trop d'inondations. On se situe entre 12% et 15% d'inondations par ensemble de 60 années. Si on reprend les poids qui viennent d'être trouvés, et qu'on relance une instance, la production de mégawatts est moins bonnes, mais les inondations restent inférieures à 10%.

La simulation 2 diffère des autres au niveau du nombre d'inondations, comme indiqué dans le tableau 4.2. Il faudra donc être prudent lors de nos analyses sur les méthodes suivantes : tenir compte de ces différences, et aussi observer si elles se perpétuent avec les différents tests.

Il apparaît aussi qu'avec les poids initiaux égaux à 1, la politique de gestion a tendance à causer des inondations dans la ville avoisinante. De plus, si on s'attarde

Ensembles d'apports	Probabilités d'inondations
Sim 1	11.7 %
Sim 2	2.5 %
Sim 3	8.33 %
Sim 4	8.33 %
Sim 5	8.33 %
Histo	9.17 %
Ensembles d'apports (sim 1 à sim 5), poids de départ = 1	11 %
Ensembles d'apports (sim 1 à sim 5), poids de départ \neq 1	4.67 %
Ensembles d'apports (sim 1 à sim 5) et apports Histo	8.05 %

TABLE 4.2 – Inondations moyennes pour la méthode 3

sur les apports historiques, là encore, les inondations sont trop élevées par rapport à la limite que nous nous sommes imposée. Avec les poids modifiés, les résultats sont beaucoup plus intéressants car ils respectent notre limite d'inondation. Nous obtenons, dès lors, notre première référence à fin de comparaison avec les méthodes suivantes. Par contre, il reste toujours une amélioration à apporter, qui est le nombre de fois ou le réservoir atteint son volume minimal. En effet, à partir du moment où 5 années (ou plus) possèdent au moins une période avec le réservoir au volume minimal, la politique devient difficilement viable. Dans cette méthode, le critère n'est pas respecté.

4.3.4 Méthode 4

La quatrième méthode reprend les bases de la méthode 2, à savoir, 10 groupes de 5 scénarios, mais avec un volume initial aléatoire à chaque instance (toujours compris entre 60% et 90% du volume maximal.)

Résultats méthode 4

Pour le premier test, l'algorithme a tendance à vider beaucoup trop rapidement le réservoir, ce qui cause des volumes trop faibles et réduit considérablement les mégawatts produits. Pour le deuxième test, les résultats sont identiques à ceux du

premier test de la méthode 3, d'où la raison de leur absence dans le tableau précédent. À ce stade, la meilleure méthode reste la numéro 3. À noter aussi que le critère en lien avec le volume minimal du réservoir n'est pas respecté ici non plus.

4.3.5 Méthode 5

La cinquième méthode propose cette fois de reprendre la méthode 2, mais d'utiliser un volume de départ aléatoire à chaque groupe de scénarios et, non plus uniquement lors de la première instance. Ainsi, le modèle est entraîné pour différents volumes de départ et la méthode est testée pour 7 scénarios.

Ensembles d'apports	Probabilités d'inondations
Sim 1	11,67 %
Sim 2	1,67 %
Sim 3	10 %
Sim 4	10 %
Sim 5	10 %
Histo	11,67 %

TABLE 4.3 – Probabilités d'inondations pour la méthode 5

Résultats méthodes 5

Pour les optimisations avec les poids égaux à 1, les résultats ne sont pas concluants car l'action proposée est toujours la même. Par contre, en réutilisant les poids trouvés dans la dernière optimisation, on obtient des résultats intéressants. Il existe un très léger dépassement de la limite d'inondation pour l'ensemble d'apports sim 1 et les apports historiques. En effet, on se retrouve à presque 12% de taux d'inondations. Encore une fois, le critère du volume minimal n'est pas respecté. Reste à voir si nous avons la capacité de nous rapprocher de ces bénéfiques tout en réduisant légèrement les inondations. On vérifie cela à l'aide des méthodes qui suivent.

4.3.6 Méthode 6

La sixième méthode propose d'utiliser le maximum de données aléatoires. En effet, on utilise 10 groupes de 3 scénarios qui seront toujours aléatoires. Une fois le passage effectué dans les dix groupes, et lorsque Nelder-Mead modifie les poids et relance une instance, les nouveaux groupes de scénarios seront différents des précédents. De surcroît, les volumes de départ sont eux aussi aléatoires.

Dans ce cas, l'algorithme ne converge pas, il s'arrête au bout du nombre maximal d'itérations. Nous avons tenté, à plusieurs reprises, de reprendre les poids trouvés à la fin d'une instance, mais les résultats finaux étaient beaucoup trop variables. C'est pour cette raison que nous ne rapportons pas de résultats dans cette section.

4.3.7 Méthode 7

Enfin, dans une dernière étape, appelée méthode 7, nous avons travaillé sur 4 compositions différentes :

- 1) Un volume fixe et le même groupe de 16 scénarios à chaque instance.
- 2) Un volume aléatoire et le même groupe de 16 scénarios à chaque instance.
- 3) Un volume fixe et un groupe de 16 scénarios aléatoires à chaque instance.
- 4) Un volume aléatoire et un groupe de 16 scénarios aléatoires à chaque instance.

Ce que nous appelons méthode 7 est en réalité, pour le moment, un ensemble de 4 sous-méthodes dont les résultats sont présentés ci-dessous. Pour les sous-méthodes 3) et 4), les scénarios aléatoires ne permettent pas de converger vers un optimum, et les résultats obtenus sur les différentes tentatives (c'est-à-dire en essayant à nouveau après que le code se soit arrêté en ayant atteint le nombre maximal d'itérations) donnent des résultats qui sont beaucoup trop disparates. Pour 1), le problème du volume fixe de départ limite l'utilisation des résultats hors-échantillons, puisque les volumes peuvent être très différents de celui choisis initialement. Seul 2) permet d'avoir des résultats qui semblent mieux s'adapter. C'est donc cette sous-méthode

qui sera retenue et à laquelle nous ferons références lorsque nous parlerons de méthode 7 dans toute la suite du mémoire.

Résultats méthode 7

Après différents essais sans effet notable sur les résultats, nous avons réussi à obtenir un ensemble de poids qui retourne une production de mégawatts très intéressants. Pour l'optimisation, nous avons utilisé, comme précisé ci-dessus, 16 scénarios, mais aussi des volumes aléatoires à chaque instance. Les poids de départs sont différents de 1 et sont ceux trouvés à l'optimisation précédente avec les mêmes critères et des poids de départs unitaires.

Les inondations sont en dessous de la limite de 10%, et les mégawatts sont élevés, et même plus élevés que dans toutes les autres méthodes testées. Par contre, nous n'arrivons toujours pas à réduire en même temps le nombre d'année où le réservoir atteint sa limite inférieure.

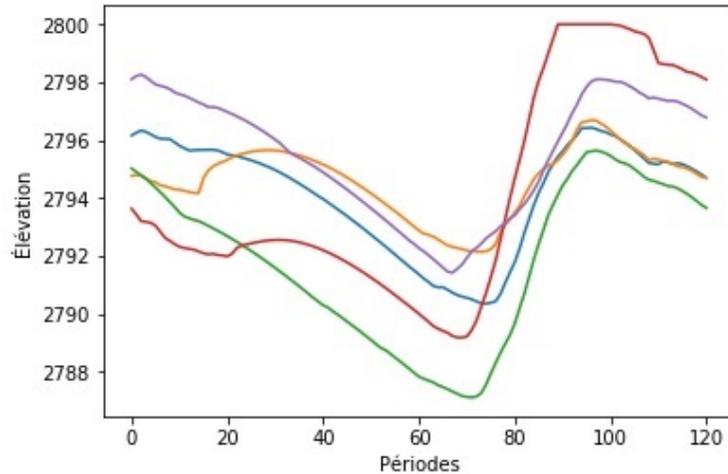


FIGURE 4.1 – Élévations régulières pour 5 scénarios pour la méthode 7

Après une description de toutes ces méthodes, il est important de faire un point sur les résultats. De plus, plusieurs remarques sont nécessaires à partir de cette étape.

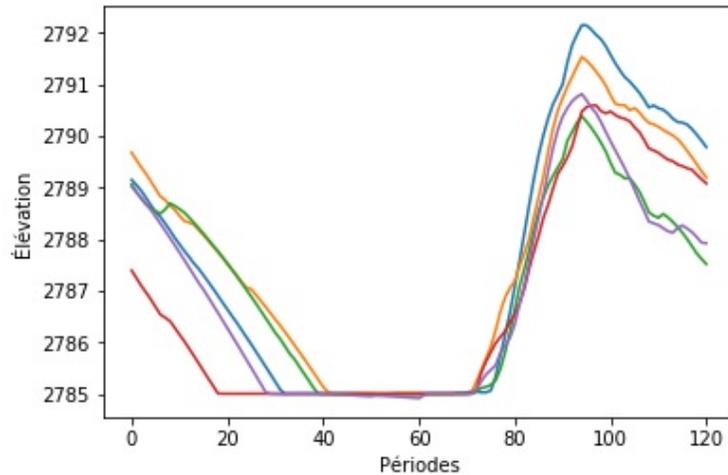


FIGURE 4.2 – Élévations irrégulières pour 5 scénarios pour la méthode 7

Résumé des méthodes utilisées :

méthode 1 : 1 groupe de 3, 5, ou 7 scénarios, volume de départ fixe à 22000 hm³

méthode 2 : 10 groupes de 5 scénarios, volume de départ fixe à 22000 hm³

méthode 3 : 1 groupe de 5 scénarios, volume aléatoire lors de la 1ère instance

méthode 4 : 10 groupes de 5 scénarios, volume aléatoire lors de la 1ère instance

méthode 5 : 10 groupes de 5 ou 7 scénarios, volume aléatoire au début de chaque groupe

méthode 6 : 10 groupes de 3 scénarios aléatoires à chaque instance, volume aléatoire au début de chaque groupe

méthode 7 : 1 groupe de 16 scénarios, volume aléatoire lors de la première instance

4.4 Interprétations des résultats et remarques supplémentaires.

Les faibles élévations dans certaines périodes laissent penser que les débits vers les turbines se réduisent considérablement dans ces moments là. C'est ce que semble

indiquer la figure 4.2. En effet, sachant que la limite inférieure du réservoir se situe à une élévation de 2785 pieds, la figure 4.2 montre bien que cette limite est atteinte plusieurs fois. Pour être plus précis, il n'y a que 6 années qui affichent des débits nuls pendant plusieurs périodes. Pour toutes les autres, malgré des débits parfois faibles, la production d'électricité est élevée et les bénéfices tout autant. Il pourrait être intéressant de reprendre des tests d'optimisations avec les scénarios comportant de faibles apports afin de vérifier le comportement de l'ensemble.

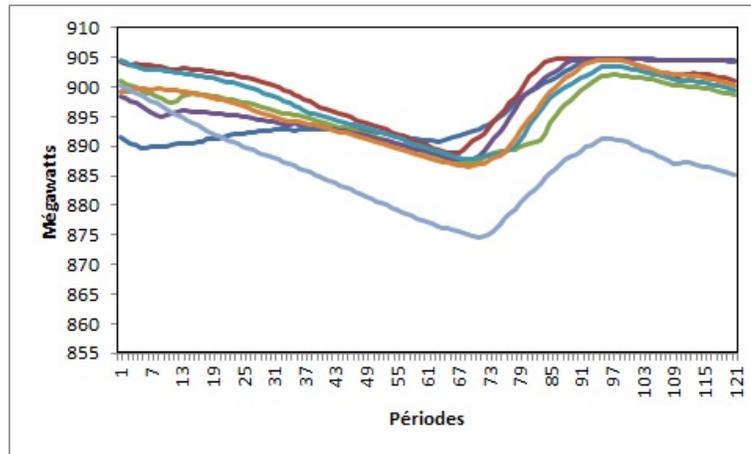


FIGURE 4.3 – Mégawatts réguliers pour la méthode 7 et simulation 1

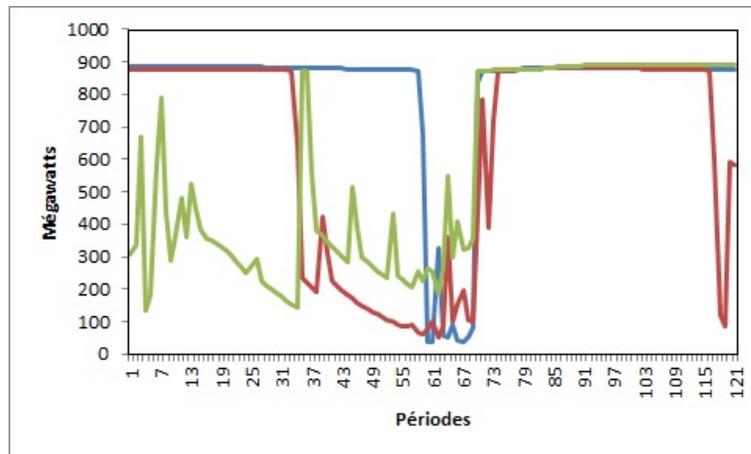


FIGURE 4.4 – Mégawatts irréguliers pour la méthode 7 et simulation 1

Les mégawatts aussi semblent être très variables selon les scénarios. La figure

4.3 présente des valeurs très élevées, c'est-à-dire bien au-dessus des 700 mégawatts nécessaires au fonctionnement de l'usine. Par contre, à la vue de la figure 4.4, l'usine n'est plus suffisamment alimentée pendant de longues périodes. Cela coïncide avec les élévations irrégulières qui empêchent d'avoir les débits nécessaires à l'atteinte des 700 mégawatts. Il ne faut pas non plus oublier de considérer le fait qu'il soit possible que plusieurs années consécutives offrent des apports faibles, ce qui cause une production faible et ce, peu importe la méthode d'optimisation.

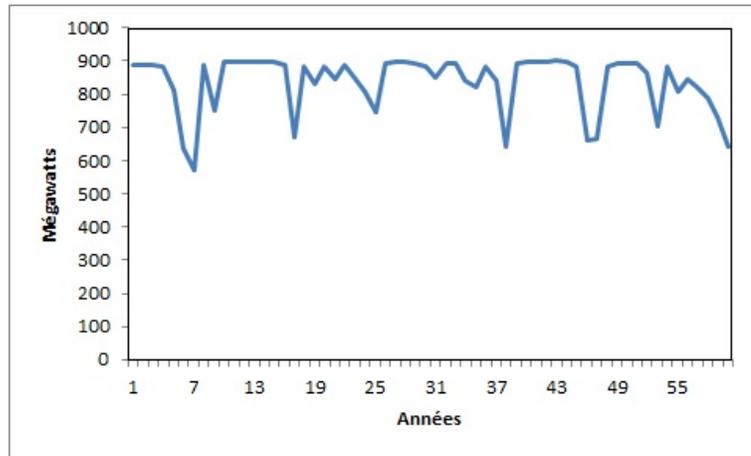


FIGURE 4.5 – Mégawatts moyen par année pour la méthode 7 et simulation 1

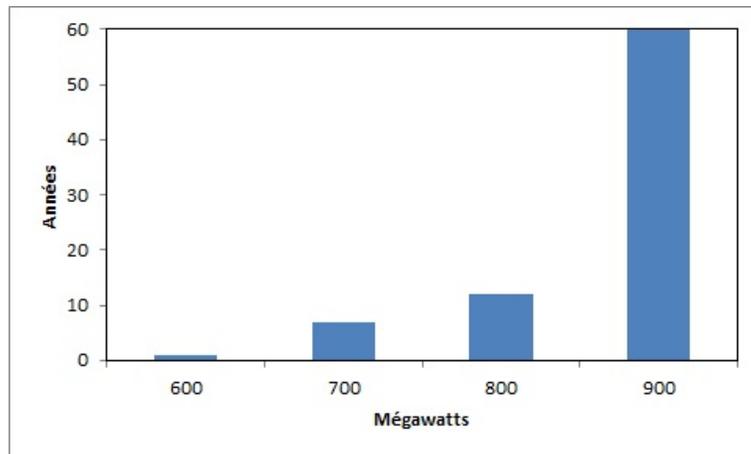


FIGURE 4.6 – Mégawatts moyen par année pour la méthode 7 et simulation 1

La figure 4.5 présente les mégawatts moyens pour les 60 années. On retrouve une

demi-douzaine de séquences où les mégawatts sont inférieurs à 700, confirmé par la figure 4.6. Cette dernière figure présente le nombre d'années où la moyenne des mégawatts est inférieure à une certaine valeur (entre 600 et 900.) On a ainsi une idée du nombre de fois où la moyenne atteint des valeurs beaucoup trop faibles. Pour les autres séquences, nous arrivons à atteindre l'objectif. Il reste qu'une politique incapable de gérer ces périodes délicates doit être améliorée.

Il reste aussi important de réaliser des tests de sensibilité pour vérifier l'impact de changements dans les couches du réseau de neurones sur les résultats. De plus, ces débits très faibles laissent tout de même place à l'amélioration, que nous présentons dans la section suivante, en modifiant les pénalités et en ajoutant des actions.

4.5 Changement du nombre d'actions disponibles et des pénalités d'inondations

Pour tenter de pallier le problème du réservoir qui se retrouve à sa limite inférieure trop souvent, nous proposons plusieurs améliorations. Tout d'abord, nous augmentons le nombre d'actions disponibles. Il est probable que la restriction à 4 possibilités limite les résultats. Nous proposons donc de passer à 8. Le tableau 4.7 en fournit la description. Les 3 premières actions ne changent pas. La 4^{ème} action est plus faible car nous n'avons pas de problème d'inondation, mais plutôt le contraire. De ce fait, nous réduisons la quantité d'eau à soutirer. Bien que cela entraîne une génération d'électricité trop faible pour faire fonctionner l'usine, nous désirons tout de même vérifier si l'algorithme l'utilise. Pour les 4 autres, nous faisons simplement une indentation de $10 \text{ m}^3/\text{s}$ jusqu'à atteindre $140 \text{ m}^3/\text{s}$.

Nous pensons que l'augmentation du nombre de variables nécessite d'en faire de même pour les neurones. Nous avons donc testé plusieurs ensembles de neurones, en débutant avec les 5 habituels, jusqu'à tenter un maximum de 30, et tout cela pour une seule couche cachée. Nous modifions aussi les pénalités sur les bénéfiques lorsqu'il

Actions	Débits Kemano	Débits Skins Lake
Action 1	110 m ³ /s	minimum légal pour la période
Action 2	148 m ³ /s	minimum légal pour la période
Action 3	148 m ³ /s	maximum légal pour la période
Action 4	90 m ³ /s	minimum légal pour la période
Action 5	100 m ³ /s	minimum légal pour la période
Action 6	120 m ³ /s	minimum légal pour la période
Action 7	130 m ³ /s	minimum légal pour la période
Action 8	140 m ³ /s	minimum légal pour la période

TABLE 4.4 – Actions modifiées du système hydroélectrique de Kemano

y a une inondation ou que le réservoir atteint sa limite inférieure. Tous les résultats se retrouvent dans la section des tests de sensibilité.

4.6 Tests de sensibilité et tests supplémentaires

Dans les différentes méthodes utilisées, le nombre de neurones reste limité à 5, et cela dans une seule couche. Nous avons vérifié si des changements dans ces deux conditions avaient une quelconque influence sur la production de mégawatts, les inondations et le nombre de fois où le réservoir atteint sa limite inférieure. Nous avons aussi ajouté quelques tests supplémentaires pour vérifier si de nouvelles modifications pouvaient avoir elles aussi un effet sur la production et les contraintes.

Changement du nombre de neurones

La première modification qui semble la plus logique est l'augmentation du nombre de neurones dans le réseau. Nous avons commencé avec 5 neurones, puis nous avons augmenté ce nombre au fur et à mesure, et ce jusqu'à 30. À nouveau, les inondations étaient trop élevées, ce qui ne correspondait pas à nos attentes.

Changement du nombre de couche

Il est maintenant utile de vérifier si une augmentation du nombre de couches cachées permet d'obtenir une meilleure production de mégawatts lors de l'optimisation. Ici, encore, les résultats ne sont pas très intéressants puisque les inondations sont de nouveaux trop élevées.

Ajout d'une troisième couche cachée

Nous avons utilisé l'algorithme pour optimiser en premier un très faible nombre de scénarios, puis augmenté au fur et à mesure en partant des derniers poids trouvés. Lors de la première optimisation, nous partions avec des poids de 1 (comme dans les méthodes précédentes.) Malheureusement, il ne semble pas que l'ajout d'une nouvelle couche offre de meilleurs résultats

Ajout de pénalités sur les bénéfices pour des volumes égaux à la limite inférieure du réservoir

L'ajout d'une pénalité dans ce cas semble logique puisque nous sommes déjà capables de limiter les inondations. Le problème qui survient est le suivant : soit la pénalité est trop faible, et le volume reste trop faible, soit elle devient trop élevée, et on se retrouve avec des situations d'inondations sévères.

Changement des bénéfices selon le nombre de mégawatts obtenus

Nous avons effectué plusieurs tests dans cette section. Tout d'abord, nous avons augmenté les bénéfices provenant de la production d'électricité. Nous les avons aussi diminués pour en étudier les conséquences. Enfin, nous les avons aussi fait disparaître pour tenter de voir à quel point l'algorithme est capable d'éviter les deux situations critiques que nous essayons de corriger. Une fois encore, les résultats différaient légèrement mais la conclusion restait la même : nous n'arrivons pas à corriger les deux situations en même temps.

Changement des groupes de scénarios

On a ici un mélange de multiples scénarios testés. Un peu comme dans les différentes méthodes étudiées au préalable, on change notre groupe de scénarios. Pour les différents essais, nous avons choisi entre 3 et 60 scénarios, parfois dans l'ordre, parfois dans un ordre aléatoire. Pour l'optimisation, nous avons aussi utilisé plusieurs ensembles de simulation dont celui des apports historiques. Une nouvelle fois, les résultats n'offrent pas de faire coïncider les deux critères en même temps.

Relance de plusieurs optimisations à partir des derniers poids obtenus

Une autre façon de tenter d'améliorer les résultats consiste à relancer plusieurs fois la même méthode avec les mêmes paramètres en utilisant cette fois les poids trouvés dans la dernière optimisation. Plus précisément, à la fin d'une optimisation, nous obtenons les derniers poids transmis par Nelder-Mead, comme à chaque fois. Cependant, au lieu de s'arrêter, on reprend les dits poids, et on recommence l'optimisation avec ceux-ci. Et on réitère ce processus jusqu'à ne plus voir d'amélioration. Dans ce cas là, on retrouvait une légère amélioration avec 2 ou 3 relances. Les conditions n'étaient toujours pas complètement respectées, mais si nous trouvons un moyen d'y arriver, cette dernière section sera utile pour améliorer les résultats.

4.6.1 Utilisation de poids aléatoires

Cette section présente un test supplémentaire qui apporte de nouveaux résultats intéressants au niveau des critères d'inondations et de respect de la limite inférieure du réservoir : l'utilisation de poids aléatoires. Nous avons d'abord générés 300 000 ensembles de poids aléatoires. Nous avons vérifié pour chacun d'entre eux la production de mégawatts qu'ils offraient et le respect des contraintes lorsqu'ils étaient utilisés tels quels dans le réseau de neurones, c'est-à-dire sans optimisation. Ensuite, nous avons choisis l'ensemble qui respectait toutes les contraintes et qui offraient la meilleure production de mégawatts, et nous l'avons utilisé comme poids de dé-

part pour une optimisation avec Nelder-Mead. Cette optimisation se servant des meilleurs poids aléatoires comme poids de départ utilisait un groupe de 16 scénarios et un volume de départ aléatoire lors de la première instance. La génération des poids aléatoires se fait avec une loi uniforme comprise entre 0 et 1. Nous étudions en premier l'évolution de la valeur de la fonction objectif selon les poids aléatoires proposés. Dans le graphique 4.7, les valeurs sur l'axe des ordonnées ne prennent en compte que la dernière meilleure valeur enregistrée.

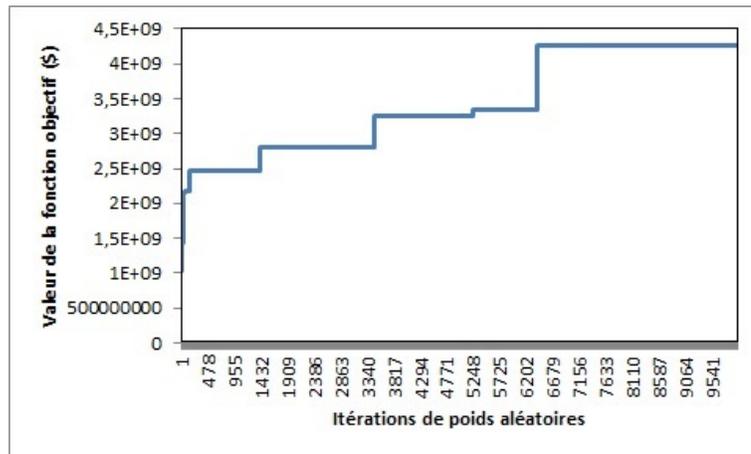


FIGURE 4.7 – Évolution de la valeur de la fonction objectif pour les 10 000 premières itérations de poids aléatoires

L'étape qui suit est alors de comparer nos résultats hors-échantillons entre les meilleurs poids obtenus grâce à Nelder-Mead en utilisant la méthode 7 (notés Poids post Nelder-Mead dans les tables) et une optimisation faite avec Nelder-Mead à partir des meilleurs poids obtenus aléatoirement (notés Poids initiaux "meilleur aléa" dans les tables). Nous allons pour cela comparer les mégawatts produits ainsi que le respect des critères (inondations et limites du réservoir.)

Les premières observations semblent montrer que les périodes où l'usine est sous alimentée sont moins élevées avec la méthode 7. Le tableau 4.5 semble aussi indiquer que la puissance moyenne est beaucoup plus élevée dans cette même méthode. Par contre, avec les meilleurs poids aléatoires, il n'arrive jamais que le niveau du réservoir se retrouve à sa limite inférieure, ce qui était un des gros problèmes avec l'autre

Ensembles d'apports	Poids post Nelder-Mead	Poids initiaux "meilleur aléa"
Sim 1	92,81 %	76,97 %
Sim 2	91,13 %	79,09 %
Sim 3	91,51 %	76,56 %
Sim 4	92,94 %	77,31 %
Sim 5	89,48 %	76,29 %
Histo	91,78 %	76,71 %

TABLE 4.5 – Pourcentage moyen des mégawatts en fonction de la puissance maximale atteignable pour les meilleurs poids obtenus avec Nelder-Mead grâce à la méthode 7 et les meilleurs poids aléatoires optimisés par Nelder-Mead

Ensembles d'apports	Poids post Nelder-Mead	Poids initiaux "meilleur aléa"
Sim 1	13,33 %	63,33 %
Sim 2	16,67 %	71,67 %
Sim 3	13,33 %	70,00 %
Sim 4	15,00 %	55,00 %
Sim 5	18,33 %	66,67 %
Histo	16,67 %	66,67 %

TABLE 4.6 – Pourcentage d'années avec une production de mégawatts inférieure à 700 pour les meilleurs poids obtenus avec Nelder-Mead en utilisant la méthode 7 et les meilleurs poids aléatoires optimisés par Nelder-Mead

méthode. À l'aide du tableau 4.6, on remarque que le nombre d'années où la production de mégawatts est inférieure à 700 mégawatts est beaucoup plus faible dans la méthode 7. Pour l'autre ensemble, on voit bien que la production est beaucoup trop faible pour fournir suffisamment l'usine. Comparons à présent les inondations à l'aide du tableau 4.7.

Ensembles d'apports	Poids post Nelder-Mead	Poids initiaux "meilleur aléa"
Sim 1	10,00 %	0,00 %
Sim 2	3,33 %	0,00 %
Sim 3	8,33 %	0,00 %
Sim 4	10,00 %	3,33 %
Sim 5	8,33 %	0,00 %
Histo	10,00 %	5,00 %

TABLE 4.7 – Pourcentage d'années en inondations pour les meilleurs poids obtenus avec Nelder-Mead en utilisant la méthode 7 et les meilleurs poids aléatoires optimisés par Nelder-Mead

Ensembles d'apports	Poids post Nelder-Mead	Poids initiaux "meilleur aléa"
Sim 1	16,67 %	0,00 %
Sim 2	15,00 %	1,67 %
Sim 3	26,67 %	1,67 %
Sim 4	18,33 %	3,33 %
Sim 5	36,67 %	5,00 %
Histo	26,67 %	0,00 %

TABLE 4.8 – Pourcentage d'années lors desquelles le réservoir atteint sa limite inférieure pour les meilleurs poids obtenus avec Nelder-Mead en utilisant la méthode 7 et les meilleurs poids aléatoires optimisés par Nelder-Mead

Eu utilisant les poids aléatoires comme points de départ pour Nelder-Mead, nous arrivons à respecter toutes les contraintes (inondation et limite inférieure du réservoir.) Il existe toujours quelques scénarios où le réservoir atteint sa limite inférieure ou produit des inondations, comme indiqué sur le tableau 4.8, mais dans un pourcentage inférieur à 10 %.

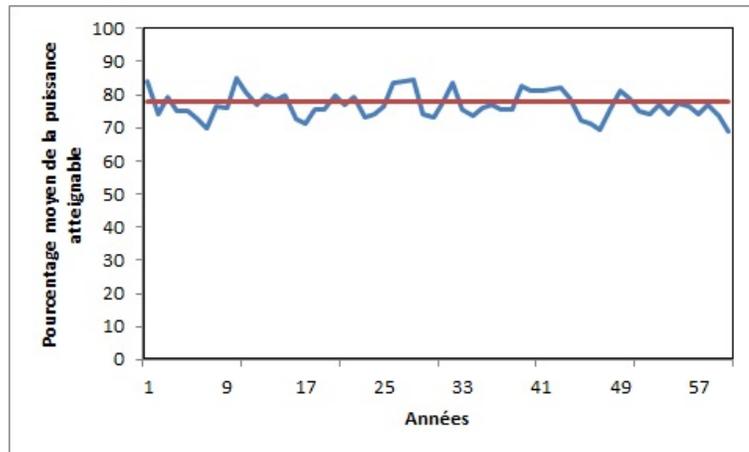


FIGURE 4.8 – Mégawatts moyen en pourcentage de la capacité atteignable par années avec l'ensemble d'apport sim 1 avec les poids obtenus de Nelder-Mead en partant de poids aléatoires

Le graphique 4.8 présente les mégawatts moyens en pourcentage de la capacité atteignable pour chaque années lors de tests hors-échantillons sur l'ensemble simulé 1. Il arrive qu'on se retrouve sous la barre des 700 mégawatts (environ 77%) requis

pour le fonctionnement de l'usine. Si on utilise les apports historiques, on obtient le graphique 4.9 qui fournit sensiblement les mêmes résultats.

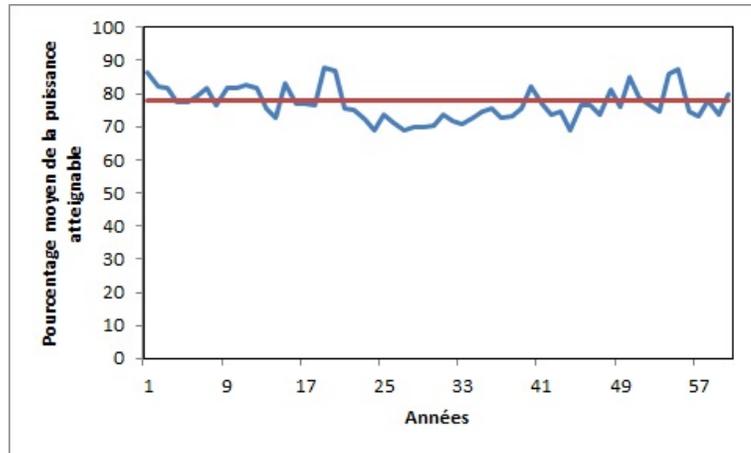


FIGURE 4.9 – Mégawatts moyen en pourcentage de la capacité atteignable par années avec les apports historiques avec les poids obtenus de Nelder-Mead en partant de poids aléatoires

4.7 Observations sur les apports et la politique

Avant de passer aux conclusions, nous passons en revue quelques observations et de nouvelles pistes de réflexion qui les accompagnent.

4.7.1 Variation des apports simulés

Les apports simulés pour chaque scénario sont relativement similaires. Si l'on regarde la variation de ces apports selon les années, on voit bien une saisonnalité, avec un faible niveau entre les périodes 30 et 60, et une forte augmentation juste après (due à la fonte des neiges.) Par contre, dans les premières périodes, les différentes années sont parfois bien différentes, comme nous pouvons le constater dans la figure 4.11. La figure 4.10 donne une vision encore plus précise de cette différence et cela pour 60 années. Cette disparité provient sûrement du fait que les saisons peuvent être très différentes selon les années. Il est possible que la fonte des neiges survienne plus tôt (ou plus tard), sur une longue période, ou très rapidement, que les pluies soient plus ou moins abondantes, etc ... C'est donc ce caractère stochastique qui pourrait expliquer ces grandes variations.

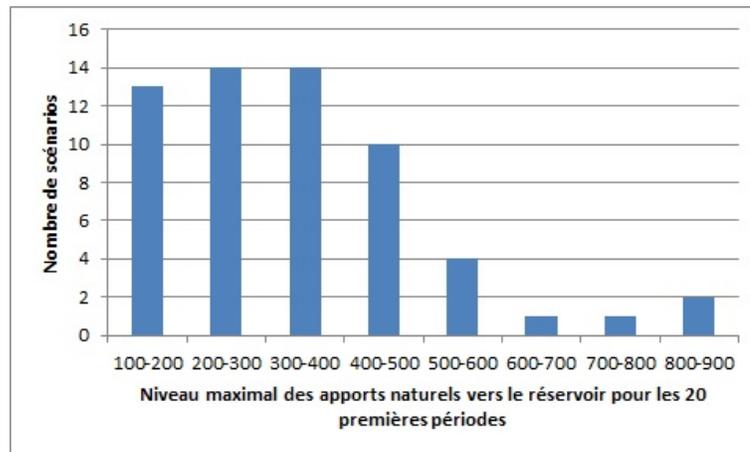


FIGURE 4.10 – Nombre de scénarios avec l'apport naturel maximum pour les 20 premières périodes

Il est évident que la plupart des scénarios (53 au total) ont des apports in-

férieurs à $600\text{m}^3/\text{s}$ pour les 20 premières périodes, mais il existe un écart non négligeable entre 100 et $600\text{ m}^3/\text{s}$. On peut alors comprendre que selon les scénarios choisis lors de l'optimisation, le résultat sera différent. La question est alors de savoir lesquels choisir. Il est peu probable que la bonne solution soit d'utiliser des scénarios extrêmes, puisque les actions qui en découleraient ne serviraient que dans un faible pourcentage de périodes. Il faut donc utiliser un mélange, mais la question reste de savoir dans quel ordre et jusqu'à quel point les scénarios avec les apports les plus faibles ou les plus élevés doivent être inclus dans l'optimisation.

4.7.2 Variation des actions utilisées par le réseau de neurones lors des phases d'apports élevés

Un point important pour vérifier si notre optimisation fonctionne correctement est d'étudier le choix des actions de notre réseau de neurones lors des situations où les apports sont très élevés. Il est logique de penser que dans de tels cas, les actions proposées utilisent des débits plus élevés et plus tôt dans l'année pour limiter les risques d'inondation. Nous allons donc travailler sur les périodes 60 à 100 puisque c'est à ce moment que les apports sont les plus élevés, comme le montre bien la figure 4.11.

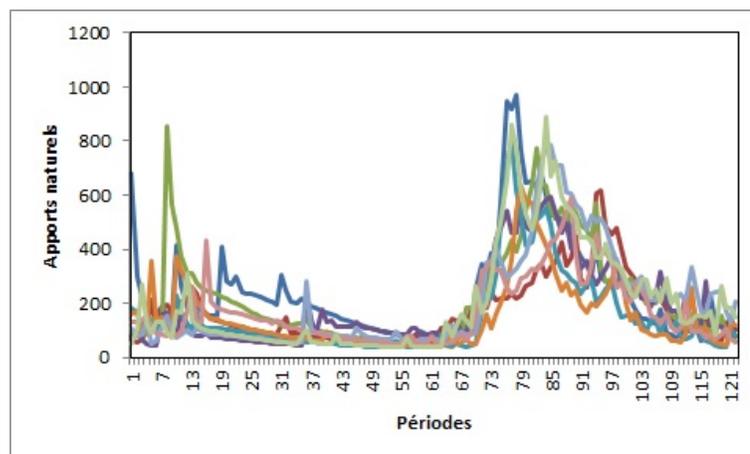


FIGURE 4.11 – Apports naturels pour plusieurs scénarios

Nous avons vérifié avec la méthode qui donnait les meilleurs résultats, et voici ce que nous avons obtenu. Tout d'abord, il faut bien comprendre que les actions proposées vont, dans une même année, rester les mêmes pendant un grand nombre de périodes. Nous nous attendons donc à ce type de résultat, et c'est effectivement ce qui arrive. Par contre, même lors du passage dans les périodes abondantes, les décisions ne changent pas (ou très peu.) Plusieurs explications peuvent être avancées. Premièrement, il est tout à fait possible que la meilleure solution soit en effet de ne pas changer de décision. Puisque les débordements ne se produisent que dans 10% des cas, il n'est pas exclu que maintenir cette décision offre de bons résultats. De plus, en lien avec la section précédente et les apports très variables pour les premières périodes, et selon les scénarios choisis pour l'optimisation, on peut supposer que le réseau adapte ses décisions en s'attendant à ce que les débits de début d'année suivante soit très élevés (donc vidange rapide du réservoir) ou très faibles (donc maintien d'un volume élevé.) Concrètement, si lors de l'optimisation, les apports au début de l'année sont très élevés, l'algorithme devrait faire en sorte d'adopter une décision qui vide rapidement le réservoir à la fin de l'année précédente pour éviter les débordements. L'inverse devrait être vrai aussi, c'est-à-dire dans le cas d'apports faibles en début d'année. Là encore, il serait pertinent de tester de nombreux groupes pour sélectionner ceux qui offrent des résultats optimaux.

Conclusion

Lors de cette recherche, nous avons utilisé un réseau de neurones et un algorithme de Nelder-Mead pour optimiser la production de mégawatts d'un système hydroélectrique. Pour cela, nous avons utilisés des apports d'eau simulés et historiques ainsi que différents ensembles de scénarios pour notre apprentissage. Les réseaux de neurones ont déjà été utilisé dans le domaine de la gestion hydroélectrique, et ce pour tenter de limiter les pertes liées aux périodes extrêmes. Notre premier objectif était de vérifier si l'utilisation des réseaux de neurones permettait une amélioration de la production d'électricité par rapport aux méthodes utilisées actuellement, tout en respectant les contraintes imposées au système. De plus, nous désirions tester la sensibilité du système par rapport à des changement au sein même du réseau de neurones.

En ce qui a trait à la production de Mégawatts, les moyennes sont identiques à celles trouvées avec les méthodes de programmation dynamique actuelles. Nous n'avons par contre pas réussi à limiter les taux d'inondations, de même que les années où le réservoir atteignait sa limite inférieure.

La première amélioration possible est l'augmentation du nombre de politiques proposées à l'administrateur. Elles étaient initialement de trois, mais nous les avons augmentées à huit pour tenter de proposer plus de solutions au réseau. Puisque les débits possibles vers Kemano sont entre 80 et 170 m³/s, on pourrait faire en sorte que chaque politique augmente d'un m³/s. Le réseau aurait alors beaucoup plus de possibilités. Évidemment, bien d'autres discrétisations sont possibles. De plus, on

pourrait aussi ajouter des entrées au réseau pour plus de précision. Par exemple, la quantité de neige accumulée, les précipitations, ou tout ce qui est jugé pertinent. Enfin, une dernière possibilité est le changement de l'algorithme d'optimisation. Dans le cas où les changements précédents n'améliorent pas les résultats, un autre algorithme pourrait être bénéfique. A contrario, si les résultats sont convenables, ce changement pourrait aider à réduire la variance que l'on retrouve lorsque l'on se rapproche de l'optimum.

Bibliographie

- [1] M. Barros and al. Optimization of large-scale hydropower system operations. *Water Resources Planning Management*, Avril 2003.
- [2] R. Bellman. Dynamic programming and stochastic control processes. *Information and Control*, pages volume 1, 228–239, 1957.
- [3] A. Cancelliere and al. A neural network approach for derivating irrigation reservoir operating rules. *Water Resources Management*, pages volume 16, 71–88, Janvier 2002.
- [4] A. Castelletti, S. Galelli, M. Restelli, and R. Soncini-Sessa. Tree-based reinforcement learning for optimal water reservoir operation. *Water Resources Research*, pages volume 46, 1–19, Septembre 2010.
- [5] A. Castelletti, F. Pianosi, and M. Restelli. A multiobjective reinforcement learning approach to water resources systems operation : Pareto frontier approximation in a single run. *Water Resources Research*, pages volume 49, 3476–3486, Juin 2013.
- [6] Q. Desreumaux. *Amélioration de la représentation des processus stochastiques pour l’optimisation appliquée à la gestion des systèmes hydriques*. PhD thesis, Université de Sherbrooke, 2016.
- [7] W. A. Hall and al. Optimization of operation of a multiple-purpose reservoir by

- dynamic programming. *Water Resources Research*, pages volume 4, 471–477, 1968.
- [8] J. e. a. Kelman. Sampling stochastic dynamic-programming applied to reservoir operation. *Water Resources Research*, pages volume 26, 447–454, Mars 1990.
- [9] J.-H. Lee and J. W. Labadie. Stochastic optimization of multireservoir systems via reinforcement learning. *Water Resources Research*, pages volume 43, 1–16, Novembre 2007.
- [10] B. W. Powell. *Stochastic optimization and learning : a unified framework*. Wiley-Interscience, 2018.
- [11] D. Rani and M. M. Moreira. Simulation-optimization modeling : A survey and potential application in reservoir systems operation. *Water Resources Management*, pages 1107–1138, Juillet 2009.
- [12] J. D. Rieker and J. W. Labadie. An intelligent agent for optimal river-reservoir system management. *Water Resources Research*, pages volume 48, 1–16, Septembre 2012.
- [13] R. S. Sutton and A. G. Barto. *Reinforcement Learning : An Introduction*. The MIT Press, 2018.
- [14] G. K. Young. Finding reservoir operation rules. *Journal of the Hydraulic Division*, pages volume 93, 297–321, 1967.

