Distribution de Processus Décisionnels Markoviens pour une gestion prédictive d'une ressource partagée : application aux voies navigables des Hauts-de-France dans le contexte incertain du changement climatique

Guillaume, Louis, Florent Desquesnes

IMT LILLE DOUAI | UNIVERSITÉ DE LILLE | CONSEIL RÉGIONAL

THÈSE

présentée en vue
d'obtenir le grade de

DOCTEUR

en

Discipline: Informatique, automatique

par

Guillaume DESQUESNES

DOCTORAT DE L'UNIVERSITÉ DE LILLE
DÉLIVRÉ PAR IMT LILLE DOUAI

Titre de la thèse:
Distribution de Processus Décisionnels Markoviens pour une gestion prédictive d'une ressource partagée: Application aux voies navigables des Hauts-de-France dans le contexte incertain du changement climatique

Soutenue le 23 octobre 2018 devant le jury d'examen:

Présidente	Nathalie Peyrard	Directrice de recherche	INRA Toulouse
Rapporteur	Olivier Buffet	Chargé de recherche	INRIA / LORIA
Rapporteur	Joseba Quevedo	Professeur	UPC Barcelone
Membre	Laëtitia Matignon	Maître de conférences	Université de Lyon
Membre	François Pérès	Professeur	INP Toulouse - ENIT
Encadrant	Arnaud Doniec	Maître de conférences	IMT Lille Douai
Encadrant	Guillaume Lozenguez	Maître de conférences	IMT Lille Douai
Directeur de thèse	Éric Duviella	Professeur	IMT Lille Douai

Laboratoire d'accueil: Département ou Unité de Recherche Informatique et Automatique de IMT Lille Douai

École Doctorale SPI 072 (Lille I, Lille III, Artois, ULCO, UVHC, Centrale Lille, IMT Lille Douai)

Remerciements

Les travaux de recherche présentés dans ce manuscrit ont été initiés par Éric Duviella, Arnaud Doniec et Guillaume Lozenguez qui m'ont encadré durant les trois années de cette thèse. Je les remercie pour leur investissement et leurs conseils avisés qui m'ont permis de mener à bien ces trois années de doctorat. J'espère avoir l'opportunité de poursuivre cette collaboration dans le futur et de continuer à profiter de leur expérience.

Je tiens à remercier Olivier Buffet, Laëtitia Matignon, François Pérès, Nathalie Peyrard et Joseba Quevedo d'avoir accepté de faire partie de mon jury. Je remercie tout particulièrement Olivier Buffet et Joseba Quevedo pour avoir accepté de rapporter ma thèse. Je les remercie également d'avoir consacré de leur temps à la lecture de mon manuscrit.

Je remercie également le conseil régional des Hauts-de-France d'avoir financé cette thèse et les Voies navigables de France pour nous avoir fourni les données nécessaires pour modéliser des réseaux réels.

Je remercie tous les membres du DIA, permanents, doctorants, post-doctorants ainsi que l'équipe administrative pour leur accueil au sein du département durant les trois années de ma thèse. Je remercie tout particulièrement Houda, Pau, Jonathan, Debora ainsi que tous les doctorants de l'équipe pour les moments de détentes que nous avons partagés ensemble.

Merci finalement à tous mes proches, mes amis et ma famille qui ont été présents et m'ont soutenu durant ces trois années.

Table des matières

Table des figures

Liste des tableaux

Liste des publications

- Desquesnes, G., Lozenguez, G., Doniec, A., and Duviella, É. (2018b). Large Markov decision processes based management strategy of inland waterways in uncertain context. In *Advances in Hydroinformatics*, pages 3–20. Springer
- Desquesnes, G., Alves, D., Lozenguez, G., Doniec, A., and Duviella, E. (2018a). Simulation architecture based on distributed MDP for inland waterway management. In *13th International Conference on Hydroinformatics, 2018*
- Desquesnes, G., Lozenguez, G., Doniec, A., and Duviella, E. (2017b). Coordination distribuée et hors-ligne de planifications locales. In *Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite de systèmes (JFPDA 2017)*
- Desquesnes, G., Lozenguez, G., Doniec, A., and Duviella, É. (2017c). Distributed MDP for water resources planning and management in inland waterways. *IFAC-PapersOnLine*, 50(1):6576–6581
- Desquesnes, G., Lozenguez, G., Arnaud, D., and Duviella, E. (2017a). Vers une distribution des MDP à grande échelle: étude de cas des voies navigables. *Revue des Sciences et Technologies de l'Information-Série RIA: Revue d'Intelligence Artificielle*, 31(1-2):183–205
- Desquesnes, G., Lozenguez, G., Doniec, A., and Duviella, É. (2016c). Planning large systems with MDPs: case study of inland waterways supervision. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 5(4):71–84
- Desquesnes, G., Lozenguez, G., Doniec, A., and Duviella, É. (2016b). MDP à grande échelle: étude de cas des voies navigables. In *Journées Applications Pratiques en Intelligence Artificielle (APIA) pendant le vingtième congrès national sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'16)*
- Desquesnes, G., Lozenguez, G., Doniec, A., and Duviella, E. (2016a). Dealing with large MDPs, case study of waterway networks supervision. In *Advances in Practical Applications of Scalable Multi-agent Systems. The PAAMS Collection*, pages 48–59. Springer

- Desquesnes, G., Nouasse, H., Lozenguez, G., Doniec, A., and Duviella, E. (2016d). A global approach for investigating resilience in inland navigation network dealing with climate change context. *Procedia Engineering*, 154:718–725
- Segovia, P., Desquesnes, G., Doniec, A., Duviella, E., Lozenguez, G., Nejjari, F., Puig, V., and Rajaoarisoa, L. (2018a). Management tools to study and to deal with effects of climate change on inland waterways. In *TRA 2018*

Introduction générale

De nos jours, l'influence du réchauffement climatique est de plus en plus visible. Selon le GIEC (Groupement d'experts intergouvernemental sur l'évolution du climat), les phénomènes météorologiques extrêmes à l'origine des périodes de sécheresses et d'inondations devraient devenir plus fréquents. Dans un même temps, l'utilisation du transport fluvial devient de plus en plus intéressante pour des raisons économiques et écologiques laissant envisager un accroissement du trafic fluvial. Ainsi, les effets probables du réchauffement climatique et de l'augmentation de la demande de navigation vont impacter fortement la gestion de l'eau sur les voies navigables et de ce fait rendre d'autant plus important le besoin de résilience sur ces réseaux. Des méthodes existantes permettent d'optimiser la gestion de la ressource en eau dans les voies navigables, mais nécessitent une connaissance la plus exacte possible de son évolution. Or, les effets climatiques et l'intensité du trafic sont des événements incertains. Par exemple, il est difficile de déterminer avec précision à quel instant et à quel endroit il pleuvra. Certaines solutions d'optimisation de la ressource peuvent prendre en compte des événements incertains, mais elles se heurtent à des difficultés lors du traitement de réseaux complexes de grande taille.

Dans le cadre de cette thèse, nous nous intéressons à optimiser la gestion de l'eau dans les voies navigables tout en prenant en compte les événements incertains. Cela nous a permis de définir une classe de problèmes plus génériques, à savoir : la gestion prédictive sous incertitudes d'une ressource partagée dans des réseaux complexes de grande taille. L'objectif est de déterminer une politique de gestion résiliente de cette ressource. Ce travail propose une approche utilisant la distribution de la modélisation et de l'optimisation sur plusieurs agents dans le but de permettre un passage à l'échelle de l'approche d'optimisation sous incertitudes proposée. Quant à la distribution de la planification, elle consiste à coordonner l'optimisation des différents modèles locaux dont les évolutions sont interdépendantes.

Cette thèse s'articule autour de cinq chapitres. Un premier chapitre présente la problématique de gestion sous incertitudes de la ressource en eau dans les réseaux de voies navigables. Dans un futur proche, cette gestion sera très impactée par les effets probables du changement climatique et par les modifications des règles de navigation. De ce fait, une adaptation de la gestion devient nécessaire. Différentes approches automatiques existent et ont été appliquées à la gestion de l'eau sur ce type de réseau hydrographique. Cependant, à notre connaissance, ces approches ne permettent pas de traiter à la fois des problèmes de tailles conséquentes et les incertitudes

affectant le réseau.

Le second chapitre est consacré à la formalisation et la modélisation de problèmes probabilistes afin de permettre l'obtention de politique de gestion sous incertitudes. Dans un premier temps, nous présentons le formalisme de modélisation des processus décisionnels markoviens (MDP). Ce formalisme prend en compte les incertitudes de façon à optimiser une solution en considérant l'ensemble des évolutions possibles du système. Cependant, les MDPs ne peuvent pas représenter des problèmes réalistes de taille conséquente à cause d'une explosion combinatoire du modèle et de sa résolution. Différentes approches permettant d'étendre les capacités de modélisation ou la taille des problèmes traitables sont introduites tels que les MDPs factorisés et les MDPs décomposés. Ces approches exploitent des caractéristiques particulières de ces problèmes, de façon centralisée ou décentralisée.

Le troisième chapitre propose une définition formelle de la classe de problèmes traitée dans cette thèse : la gestion prédictive sous incertitudes d'une ressource partagée sur un réseau de grande taille. Une définition de la modélisation de tels problèmes, avec un domaine de fonctionnement continu, par des MDPs est proposée permettant ainsi la prise en compte des incertitudes. Une discrétisation des états et des actions est alors utilisée. L'impact de cette discrétisation sur l'évolution du réseau est pris en compte afin de le minimiser. Cette modélisation est illustrée sur un exemple de gestion de la ressource eau dans les réseaux de voies navigables. L'applicabilité à cette classe de problèmes des différentes approches de la littérature introduites dans le chapitre précédent est discutée. Ces approches étant inadaptées à la résolution de la problématique de cette thèse, une nouvelle approche est donc proposée.

Un nouvel algorithme intitulé « coordination hors-ligne de planifications locales » a été proposé. Il met en œuvre une distribution à la fois de la modélisation et de l'optimisation sur plusieurs agents. Il est détaillé dans le quatrième chapitre. La distribution a pour but de faciliter le passage à l'échelle de la modélisation. Pour cela, les capacités de contrôle du système traité sont réparties entre les agents. Le modèle de chaque agent est défini uniquement par le sous-réseau qu'il affecte. De ce fait, les agents se coordonnent de proche en proche pour trouver une solution à leur problème local grâce à l'utilisation de MDP. Cette approche est illustrée sur un problème académique afin de bien visualiser son fonctionnement. Une comparaison des politiques produites par cette approche avec des MDPs centralisés est proposée pour évaluer la qualité relative des solutions obtenues ainsi que les ressources de calcul nécessaires à leurs obtentions. La méthodologie proposée nécessite une décomposition du problème en agents. Pour cela un algorithme heuristique de décomposition est également décrit.

Enfin, le cinquième chapitre s'intéresse à l'évaluation de l'approche d'optimisation distribuée proposée sur différents réseaux de voies navigables, tout d'abord factices puis réalistes. Ces cas d'études permettent de valider nos contributions. Ces expérimentations évaluent l'approche proposée sur différents critères tels que la qualité des solutions calculées, le passage à l'échelle en temps de calcul et en mémoire. Elles visent aussi à montrer la capacité des solutions produites à rendre résiliente la gestion des réseaux. Les calculs sont réalisés selon les conditions actuelles de

gestion, mais aussi en considérant des effets probables du changement climatique et de gestion du réseau. Les politiques de gestion ainsi calculées sont appliquées dans un simulateur simplifié des voies navigables.

Chapitre 1

Gestion adaptative de la ressource en eau pour le transport fluvial

Les travaux de cette thèse sont motivés par la problématique de gestion optimale de l'eau sous incertitudes dans les réseaux de voies navigables. Dans ce premier chapitre, nous définissons ce qu'est un réseau de voies navigables, son fonctionnement actuel et ses objectifs de gestion. Nous présentons les raisons rendant nécessaire la gestion automatique d'un tel réseau afin de le rendre plus résilient au changement climatique et à l'augmentation de la demande de navigation. Une présentation des approches existantes les plus pertinentes visant à résoudre des problèmes similaires est proposée afin de positionner nos travaux sur ce domaine d'application.

1.1 Présentation du contexte de la thèse

Un réseau de voies navigables est un réseau hydrographique, aménagé par l'homme, interagissant avec un environnement naturel. Il s'agit d'un système à grande échelle, voir figure 1.1, qui est majoritairement utilisé pour la navigation, comme le transport de marchandises à l'aide de péniches. Ce type de réseau fournit à la fois des avantages économiques et environnementaux [Mallidis et al., 2012, Mihic et al., 2011], tout en fournissant un transport discret, efficace et sûr des biens [Brand et al., 2012]. Il offre une alternative intéressante aux transports routiers et ferroviaires.

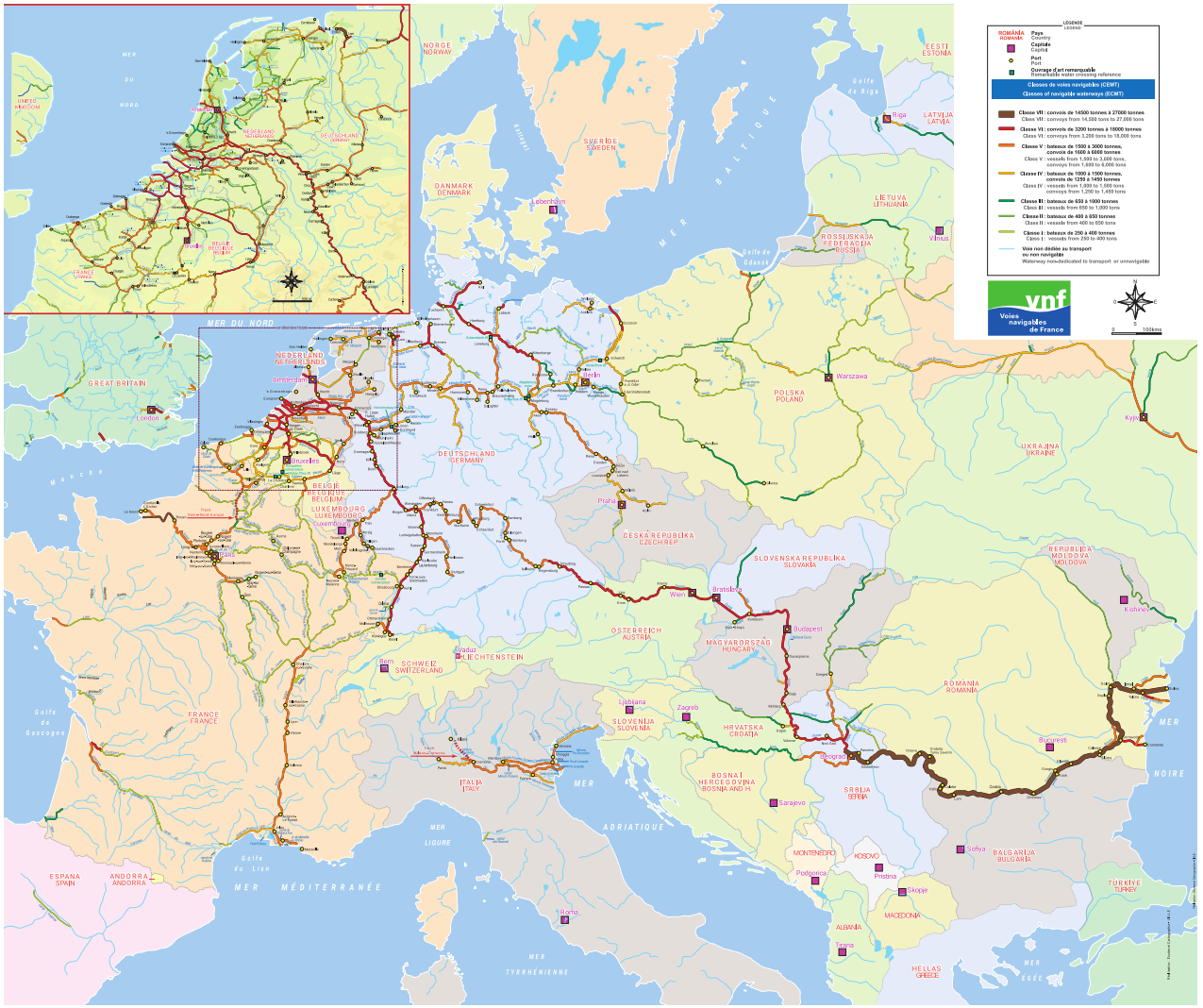

FIGURE 1.1 – Carte des réseaux de voies navigables européens

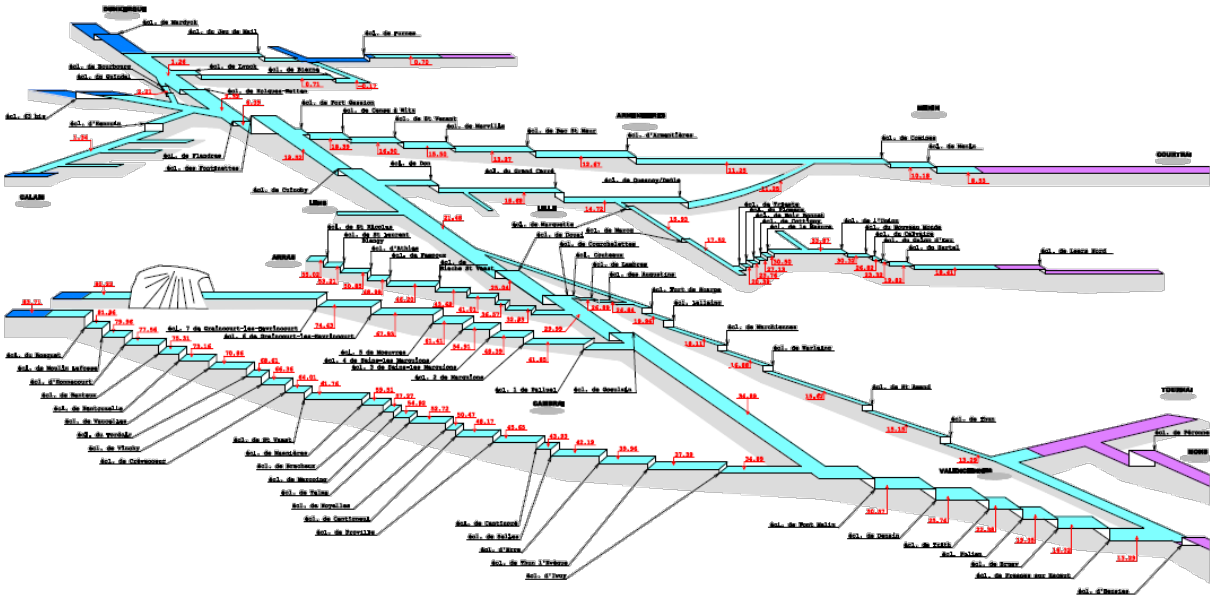

FIGURE 1.2 – Schéma du réseau de voies navigables du Nord-Pas-de-Calais

1.1.1 Réseau de voies navigables

Les réseaux de voies navigables sont généralement constitués de plusieurs biefs pour rendre possible la navigation sur des cours d'eau n'étant pas initialement navigables, ou entre des cours d'eau adjacents grâce à la construction de canaux artificiels. Un bief est une partition d'un canal ou d'une rivière canalisée dont le niveau est uniforme et suffisamment élevé pour permettre le passage des navires. Un bief est généralement délimité par deux écluses : une en amont et une en aval. Celles-ci maintiennent le niveau d'eau dans chaque bief en bloquant l'écoulement naturel de l'eau. L'utilisation des écluses pour permettre le passage de navires implique une grande consommation d'eau se comptant en milliers de mètres cubes. Afin de faciliter le trafic fluvial, il sera important de gérer efficacement la ressource en eau de ces réseaux. La figure 1.2 représente schématiquement l'ensemble des biefs composant le réseau de l'ancienne région Nord-Pas-de-Calais.

Fonctionnement du réseau

Les écluses ont pour principal objectif de permettre aux navires de changer leur élévation afin de passer d'un bief à l'autre. Une écluse, voir figure 1.3, est constituée d'au moins deux portes et d'un sas. L'ouverture d'une des portes permet d'aligner le niveau du sas sur celui du bief correspondant. Dans l'exemple de la figure 1.3, le niveau du sas est égal à celui du bief aval. Le navire peut rentrer dans le sas de l'écluse grâce à la porte aval (à gauche sur la figure).

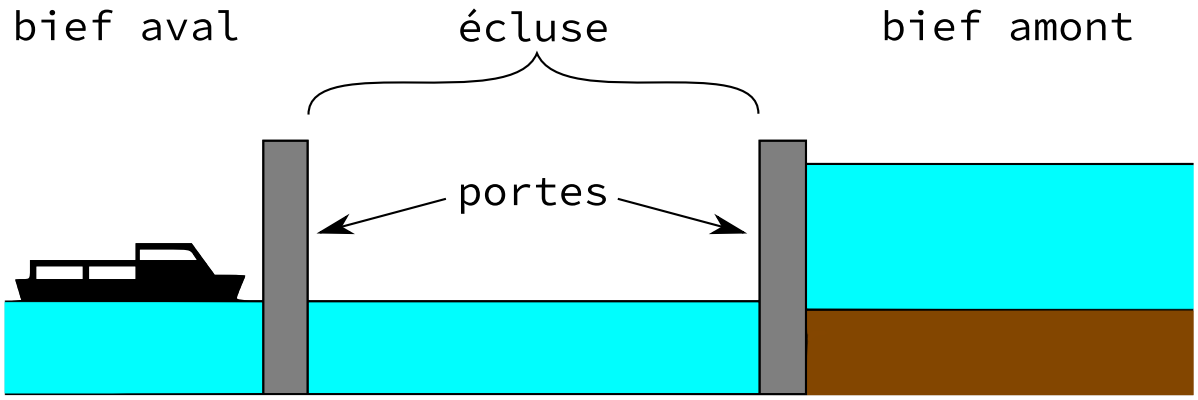

FIGURE 1.3 – Schéma d'une écluse, composée de deux portes, séparant deux biefs

En n'ouvrant que la porte amont (à droite), le niveau du sas pourra ainsi s'égaliser avec le bief amont. De cette façon, le navire pourra atteindre le bief amont, malgré la différence de niveaux entre les deux biefs.

L'utilisation d'une écluse modifie le niveau des deux biefs qu'elle connecte, puisqu'un volume d'eau est transféré d'amont en aval à chaque utilisation. Or pour que la navigation soit possible, le « rectangle de navigation » doit être maintenu, voir figure 1.4. Ce rectangle, défini pour chaque bief, est représenté par la largeur navigable du bief ainsi que les hauteurs d'eau permettant aux plus gros navires d'être à une distance raisonnable du fond et des ponts.

À partir de ce rectangle, trois niveaux sont définis :

1. le niveau haut de navigation (HNL - Highest Navigation Level) ;
2. le niveau bas de navigation (LNL - Lowest Navigation Level) ;
3. le niveau normal de navigation (NNL - Normal Navigation Level).

Les deux premiers niveaux, le HNL et le LNL, sont les niveaux correspondant respectivement aux niveaux maximal et minimal permettant la navigation du navire de plus fort tonnage autorisé sur ce bief.

Le troisième niveau, le NNL, est défini par les gestionnaires dans l'intervalle [LNL, HNL]. Il correspond au niveau idéal permettant, de façon générale, de maximiser les chances de rester dans le rectangle de navigation. Ces trois acronymes seront utilisés dans la suite du document pour parler du niveau du bief et du volume d'eau correspondant.

Gestion actuelle du réseau

Dans une situation normale, l'utilisation des écluses est la principale perturbation du niveau d'un bief. En effet, les passages entre biefs requis par la navigation impliquent un grand transfert d'eau de l'amont vers l'aval. Néanmoins, d'autres perturbations peuvent affecter le niveau

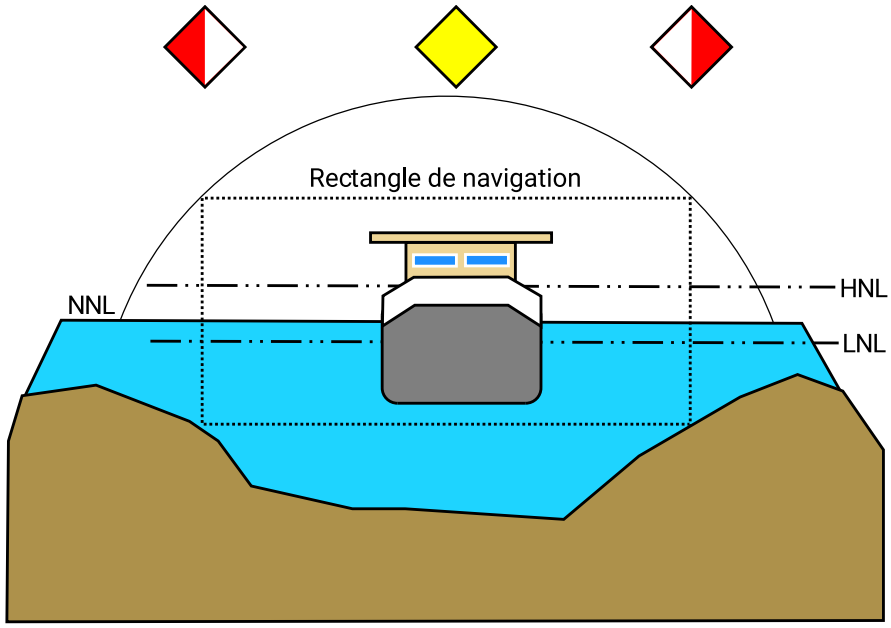

FIGURE 1.4 – Rectangle de navigation d'un bief. Le niveau haut de navigation (HNL - Highest Navigation Level) et le niveau bas de navigation (LNL - Lowest Navigation Level) sont les niveaux correspondant aux limites permettant la navigation, le niveau normal de navigation (NNL - Normal Navigation Level) est le niveau idéal pour la navigation

des biefs, par exemple : le déversement de rivières naturelles ; les événements météorologiques (e.g. pluie, sécheresse) ; l'impact humain (e.g. irrigation) ou encore les échanges avec les nappes phréatiques.

Les écluses ne sont pas dédiées au contrôle du niveau d'eau. Pour cela, de nombreux ouvrages d'art (vannes, barrages, pompes, ...) répartis sur le réseau permettent aux gestionnaires et aux opérateurs de déplacer l'eau à l'intérieur du réseau supervisé, mais aussi depuis ou vers l'extérieur, comme les rivières naturelles réseau d'un pays étranger. Les opérateurs se basent sur l'expertise accumulée au cours du temps pour maintenir les biefs qu'ils contrôlent au niveau requis, en fonction de ce qu'ils observent localement, à l'aide de ces ouvrages d'art.

En France, sur les axes principaux, le trafic fluvial peut être divisé en deux périodes : une période de « nuit » d'une durée d'environ 10 heures durant laquelle les écluses sont fermées et une période de « jour » d'environ 14 heures.

La gestion du réseau étant actuellement basée sur une expertise humaine acquise au cours du temps, cela rend difficile de projeter l'application de cette expertise à des changements importants sur le réseau.

Évolutions du réseau

Le changement climatique est étudié depuis de nombreuses années. Le Groupe d'experts Intergouvernemental sur l'Évolution du Climat (GIEC) estime qu'il impactera à la fois la température et les précipitations. Des études récentes [Beuthe et al., 2014, Boé et al., 2009, Ducharne et al., 2010] prévoient, dans un futur proche, une augmentation importante de la fréquence et de l'intensité des périodes d'inondation et de sécheresse sur les réseaux hydrographiques français et européens.

De plus, grâce à sa compétitivité économique et à ses avantages écologiques par rapport aux transports ferroviaire et routier [Brand et al., 2012], ainsi qu'une forte volonté politique, le transport fluvial est en pleine expansion. De ce fait, une hausse notable du trafic fluvial en Europe est attendue et estimée à +35% d'ici 2050 [Beuthe et al., 2012]. Pour accommoder cette augmentation de trafic, l'ouverture à la navigation continue (jour et nuit) est prévue à l'horizon 2020 en France [1].

1.1.2 Objectifs de gestion

L'augmentation du trafic fluvial combinée à un plus grand impact météorologique sur le réseau pourra montrer les limitations des stratégies de gestion actuelles. Une étude et potentiellement une redéfinition de ces stratégies deviennent indispensables afin d'optimiser la gestion de la ressource en eau dans les prochaines années. De plus, la prise en compte d'un horizon de gestion cohérent avec les capacités de prévisions météorologiques sera nécessaire afin de pouvoir anticiper les événements potentiels.

1. `http://www.nordpasdecalais.vnf.fr/IMG/pdf/Enjeux_Transport_cle1b453e.pdf` (23 février 2018)

Maintien des conditions de navigation du réseau

Un des objectifs principaux consiste à adopter une gestion globale du réseau de manière à optimiser l'ensemble de biefs plutôt que d'optimiser chaque bief de façon indépendante. Le but est donc de garantir que l'ensemble des biefs se trouvent dans leur rectangle de navigation à chaque instant, tout en minimisant l'écart global du niveau des biefs avec leur NNL.

Optimiser un ensemble de biefs sur un horizon de gestion de plusieurs jours permet une gestion plus efficace de l'écart du niveau des biefs au NNL sur la durée. Cela offre de nouvelles possibilités d'anticipation comme par exemple dégrader temporairement l'écart d'un bief à son NNL afin d'améliorer significativement celui d'un ou plusieurs autres biefs dans un futur proche.

Afin de pouvoir maintenir les conditions de navigation sur la durée, il est important de pouvoir prendre en compte les événements incertains et inconnus qui pourront affecter le réseau dans le but de rendre ce dernier résilient.

Résilience du réseau

Selon [Walker et al., 2004], la résilience d'un système est définie comme la capacité à absorber une perturbation, se réorganiser et continuer à fonctionner de la même manière qu'avant la perturbation. Dans cette thèse, il est considéré qu'un système est résilient si, dans la mesure du possible, les objectifs principaux du système sont respectés même si le maintien des objectifs secondaires est dégradé.

Dans cette thèse, une perturbation sur un réseau de voies navigables est considérée comme entièrement absorbée si celle-ci ne dégrade pas les conditions de navigation du réseau. Une absorption sera considérée comme partielle, si la dégradation de l'optimalité est minime par rapport à l'intensité de la perturbation. Par simplicité et clarté, nous utiliserons la capacité à anticiper les événements pour faire référence à l'absorption. Dans le cas d'une sécheresse de courte durée, cela reviendrait, si besoin, à monter préalablement le niveau du bief de façon à ce que la sécheresse le ramène à son niveau idéal. Si la sécheresse dans des conditions idéales implique un manque d'eau de 100 unités de volume, n'avoir qu'un manque de 20 unités après cet événement indique une absorption partielle tandis que de ne pas avoir de manque correspondrait à une absorption totale.

La réorganisation, ou récupération, d'un réseau intervient lorsque celui-ci se retrouve dans un état inattendu dû à la politique de gestion utilisée. Un système est capable de se réorganiser s'il possède un plan de gestion lui permettant de retrouver rapidement ses conditions de navigation après l'occurrence d'une perturbation imprévue. L'objectif est de minimiser le temps de récupération.

Finalement, nous insistons sur le fait que, dans ce cas, la résilience est une propriété du réseau et non d'un bief. Certains événements ne peuvent être anticipés en ne considérant que le bief affecté et requièrent donc la coordination de plusieurs biefs en amont et/ou en aval. Pour cela, il nous faut donc définir une planification permettant de coordonner la gestion de plusieurs biefs

pour optimiser le réseau qu'ils constituent, idéalement sur la base d'une connaissance statistique des événements à venir sur un horizon de plusieurs jours. Par exemple, si une période de pluie intense localisée au niveau d'un bief est prévue, il peut être préférable au préalable de réduire le volume du bief et des biefs amont afin qu'ils puissent être utilisés pour stocker de l'eau. Ainsi lorsque la perturbation arrive, le surplus d'eau qui n'aurait pas été absorbé pourra facilement être évacué vers les biefs aval.

Planification automatisée

Dans le but d'améliorer la résilience du réseau dans un futur proche, la gestion par expertise humaine pourrait être aidée, voire remplacée, par une planification automatisée du réseau. Ceci afin de pouvoir anticiper le mieux possible les événements répartis sur de larges réseaux. Pour maximiser la résilience du réseau, il faut anticiper au maximum les événements connus qui peuvent l'affecter afin de minimiser leurs impacts. Deux types d'événements sont définis pour l'anticipation :

- **les événements certains** qu'il est nécessaire d'anticiper en coordonnant le réseau sur la durée ;
- **les événements incertains** dont l'intérêt d'anticipation dépendra de leur probabilité et de leur potentielle intensité.

Dans les limites du possible, il faudra anticiper tout événement risquant d'empêcher le respect des conditions de navigation. Un troisième type d'événements peut être défini : les événements inconnus, tel que la défaillance d'un ouvrage d'art. Ces événements ne pouvant être anticipés, ils nécessiteront une politique de récupération.

Sur les réseaux de voies navigables, la majorité des perturbations sont incertaines. Il est possible d'obtenir des estimations plus ou moins précises sur le trafic et donc sur l'utilisation des écluses, néanmoins il ne s'agit que d'estimation. Le problème est identique pour les phénomènes météorologiques et les échanges avec les éléments externes au réseau inconnus (rejets sauvages, irrigation, ...) ou non gérés (rivières naturelles, réseaux étrangers, ...).

Pour cette automatisation de la gestion de la ressource en eau, un schéma général a été proposé dans [Segovia et al., 2018a] et est représenté sur la figure 1.5. Ce schéma définit les différentes caractéristiques, contraintes et objectifs de la gestion de l'eau dans les réseaux de voies navigables. Pour cela, trois blocs principaux sont définis :

- SCADA (système de contrôle et d'acquisition de données) :
  Ce bloc correspond aux observations de l'état actuel des voies navigables, grâce aux différents capteurs de niveau répartis sur le réseau, et aux modifications de ce dernier grâce aux consignes produites par les autres blocs.
- Aide à la décision :
  Des stratégies de gestion haut niveau prenant en compte l'évolution des réseaux sur de grands horizons de prédiction dans le but d'optimiser les niveaux d'eau sur le réseau. Il s'agit de la partie qui nous intéresse dans cette thèse.

— Accommodation de la commande :
À partir de ces stratégies de haut niveau, ce bloc détermine les consignes exactes à appliquer en les adaptant aux contraintes d'un horizon de gestion plus court et aux observations réelles.

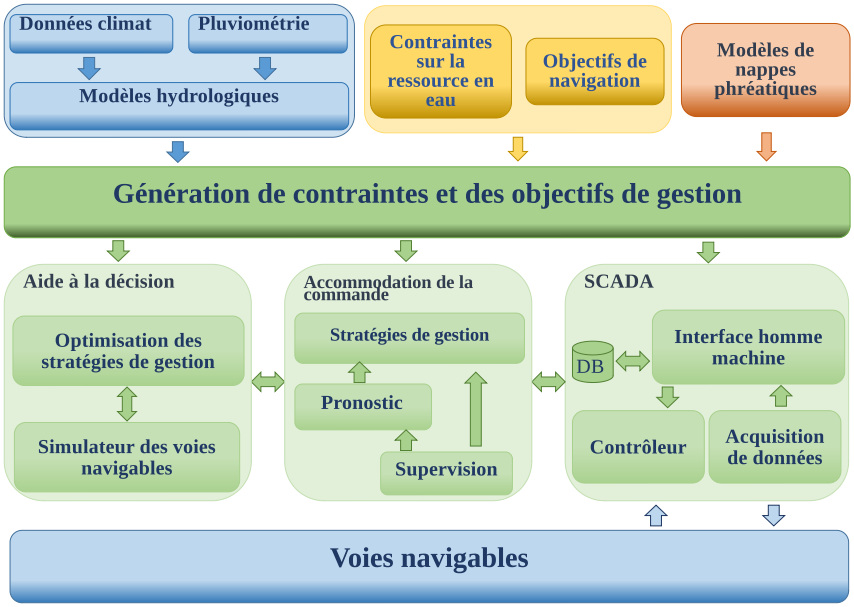

FIGURE 1.5 – Schéma général du problème

Le but de cette thèse est donc de proposer un outil de planification automatisée permettant une gestion efficiente des voies navigables capable de coordonner un réseau de plusieurs biefs pour anticiper des événements incertains tout en étant capable de restaurer rapidement les conditions de navigation en cas d'échec.

1.2 Approches existantes et connexes au sujet

Différentes approches ont été proposées pour optimiser la gestion de l'eau des voies navigables afin de les rendre résilientes. Des outils ont été développés pour mesurer ponctuellement la résilience d'un réseau pour une période de navigation donnée. D'autres approches sont définies pour absorber les perturbations futures en utilisant une connaissance *a priori* des événements qui affecteront le réseau. Enfin, la prise en compte d'événements incertains pouvant affecter un bief est utilisée afin de mitiger leurs impacts.

1.2.1 Approche par satisfaction de contrainte

L'approche proposée dans les travaux de [Nouasse et al., 2015, Nouasse et al., 2016] consiste à trouver une solution permettant de maintenir le rectangle de navigation de chaque bief au pas de temps suivant. L'hypothèse est faite que tous les déplacements d'eau, non contrôlés par un opérateur, sont connus sur l'ensemble du réseau.

Pour cela, une modélisation par graphe de flots est proposée, voir figure 1.6. Cela correspond à un graphe orienté acyclique dont les nœuds sont les biefs du réseau (NR) et les arcs correspondent aux déplacements d'eau possibles entre deux nœuds. Deux nœuds supplémentaires, une source (O) et un puits (N) sont ajoutés pour représenter les échanges d'eau avec l'extérieur du réseau. Chaque arc possède une capacité de transfert ϕ_a bornée par les volumes minimaux (l_a) et maximaux (u_a) pouvant être déplacés entre deux nœuds. Par exemple, sur la figure 1.6, l'arc reliant le nœud O à NR_i à une capacité ϕ_{Oi}. À chaque nœud est affecté un volume relatif V_i ainsi qu'un intervalle de validité reproduisant les limites du rectangle de navigation.

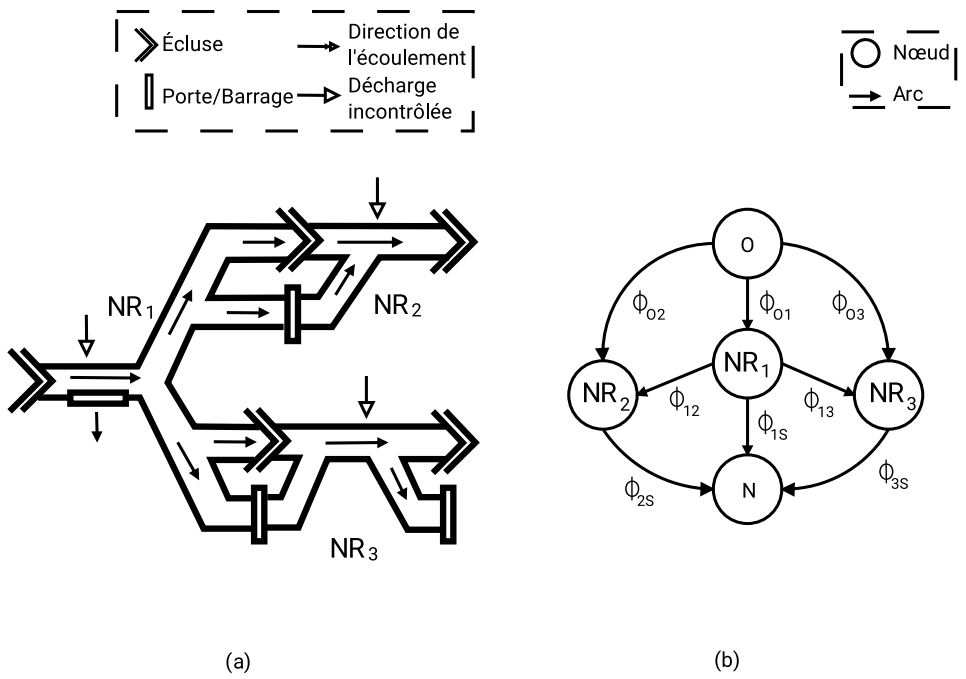

FIGURE 1.6 – Schéma et graphe de flots d'un réseau de 3 biefs ($\mathrm{NR}_{[1,\ 2,\ 3]}$). Les nœuds O et N sont respectivement les sources et les puits du graphe et les arcs correspondent aux transferts d'eau d'un nœud à l'autre.

Un problème de satisfaction de contrainte (CSP - Constraint satisfaction problem) est alors défini [Tsang, 1993]. Un CSP est composé de trois ensembles :

— X un ensemble de variables : $X = \{x_1, \ldots, x_n\}$;

— D un ensemble de domaines pour les variables de X : $d_i \in D$ le domaine de la variable $x_i \in X$;
— C un ensemble de contraintes sur les variables ; une contrainte limite les affectations possibles d'un sous-ensemble de variables.

La résolution d'un CSP consiste à trouver, si elle existe, une assignation possible de chaque variable respectant l'ensemble des contraintes.

Ici, le CSP a pour but de trouver pour chaque arc a un volume v_a à transférer respectant sa capacité de transfert ($v_a \in [l_a,u_a]$). De façon à ce que, pour un bief i, la différence des volumes entrants (v_i^+) et sortant (v_i^-) permette de conserver le réseau à l'intérieur du rectangle de navigation ($V_i + v_i^+ - v_i^- \in [V_i^{LNL},V_i^{HNL}]$). V_i^{LNL} et V_i^{HNL} sont les volumes correspondant respectivement aux LNL et HNL du bief i et V_i le volume initial du bief. Les domaines $[l_a,u_a]$ et $[V_i^{LNL},V_i^{HNL}]$ auront été précédemment discrétisés en un ensemble fini de valeurs possibles.

La résolution de ce CSP permet de trouver une solution au problème de gestion, s'il en existe une. Cette solution garantit le maintien dans le rectangle de navigation. Elle n'assure cependant pas la minimisation des écarts au NNL des biefs. De plus, l'horizon de calcul se limite à un seul pas de temps. Il s'agit de déterminer si les conditions de navigation peuvent être maintenues à partir d'un état courant du réseau en supposant que les perturbations sont connues et certaines. Les résolutions successives de CSP ne constituent pas une prise de décision séquentielle. Par exemple, si à un instant t plusieurs solutions sont possibles, le choix se fera au hasard et non par rapport aux états futurs ($t+1$, $t+2$, ...) qui découlent de ces solutions.

1.2.2 Approche par optimisation quadratique

[Duviella et al., 2016] propose deux approches permettant de planifier la gestion de l'eau sur un horizon de plusieurs jours. Elles se basent sur une extension de la modélisation par réseau de flots de [Nouasse et al., 2015] en y ajoutant un horizon de prédiction et la recherche d'une solution optimale. L'objectif devient donc de trouver les volumes à transférer $\phi_a(t)$ à chaque pas de temps t dans l'intervalle continu $[l_a(t),u_a(t)]$ afin de maximiser la proximité des volumes de chaque bief $V_i(t)$ envers leur NNL sur un horizon de τ pas de temps.

La première approche proposée utilise une optimisation pas à pas. Elle définit une fonction à optimiser de la forme :

$$J_V(t) = \sum_i W(|V_i^{NNL} - V_i(t)|) + \sum_a w_a(t) \times \phi_a(t), \tag{1.1}$$

où W est une fonction visant à pénaliser les écarts entre le volume actuel du bief $V_i(t)$ et le volume correspondant au NNL : V_i^{NNL}. w_a est un coût lié au déplacement d'eau. Par exemple, le pompage de l'eau aura un coût plus important qu'un apport d'eau de façon gravitaire. L'objectif consiste à minimiser les écarts au NNL ainsi que le coût global de gestion du système.

Une méthode par descente de gradient pour trouver le minimum d'une fonction non linéaire

multivariable sous contraintes, *fmincon*[2], est utilisée pour optimiser à un pas de temps la fonction $J_V(t)$ (équation 1.1), en respectant les contraintes de transfert et des rectangles de navigation. L'algorithme va déterminer la valeur optimale de $\phi_a(t) \forall a$ en fonction de l'état précédent du système ($V_i(t-1)\ \forall i$), des contraintes sur le transfert ($\phi_a(t) \in [l_a(t), u_a(t)]$). L'état initial du système $V_i(0)$ doit être connu. L'état atteint $V_i(t)$ est contraint et $V_i(t+1)$ sera cherché sans remise en cause des V_i précédents, ce qui implique une absence d'anticipation.

La seconde approche propose de planifier pour tous les pas de temps d'un horizon de gestion en une seule fois. L'objectif est donc d'optimiser la combinaison des objectifs de chaque pas de temps :

$$J_V^{\tau} = \sum_{t=1}^{\tau} J_V(t). \tag{1.2}$$

La méthode d'optimisation précédente, *fmincon*, est de nouveau utilisée, mais comme les volumes intermédiaires $V_i(t)$ ne sont pas connus *a priori*, il est nécessaire de les définir en fonction de l'état initial et des contrôles précédents :

$$V_i(t) = V_i(0) + \sum_{m=1}^{t-1} (v_i^+(m) - v_i^-(m)). \tag{1.3}$$

Ainsi, cette approche fournira une gestion optimale des conditions de navigation pour l'horizon souhaité, selon une évolution déterministe du système. Aucune incertitude n'est prise en compte. Plus récemment, dans [Duviella et al., 2018] la fonction *quadprog*[3] est utilisée à la place de *fmincon*. L'avantage de cette fonction est qu'elle permet d'obtenir une solution par résolution analytique et non par approximations successives.

1.2.3 Approche par commande prédictive

La commande prédictive (MPC - Model Predictive Control) [Camacho and Alba, 2013] est une commande très utilisée pour l'optimisation et la commande prédictive de systèmes dynamiques avec des applications tels que les systèmes d'irrigations [Prodan et al., 2017], de distribution d'eau [Wang et al., 2017] ou encore de barrages [Ficchì et al., 2015]. Elle utilise une modélisation dynamique d'un processus pour anticiper des événements déterministes. Pour cela, le modèle optimise l'assignation de variables continues sur un horizon fini H, via la minimisation d'une fonction multi-objectifs pondérée J :

$$\min_{i=1}^{H} J_{t+i|t} \qquad \text{avec } J_t = \sum_{j=1}^{N} \beta^j J_t^j, \tag{1.4}$$

où β^j est une pondération sur le $j^{\text{ième}}$ objectif. Chacun des N objectifs est soumis à un ensemble de contraintes quant aux valuations possibles des variables de contrôle.

2. `https://fr.mathworks.com/help/optim/ug/fmincon.html`
3. `https://fr.mathworks.com/help/optim/ug/quadprog.html`

Le plus souvent, seule la prédiction de l'état courant est utilisée et l'optimisation est relancée pour les états futurs. Ainsi les MPCs fournissent non seulement une capacité d'anticipation déterministe, mais aussi la possibilité de réagir *a posteriori* aux événements inattendus.

[Şahin and Morari, 2010] propose une modélisation décentralisée d'un réseau de 35 biefs en cascade, c'est-à-dire un enchaînement de biefs sans confluence ni diffluence. Chaque bief est modélisé par un MPC puis le réseau est optimisé en cascade. La gestion du bief le plus en amont est calculée en premier puis son impact sur les biefs directement en aval est communiqué à ces derniers afin qu'ils puissent optimiser leur gestion à leur tour. Cette étape sera répétée jusqu'à ce que l'ensemble du réseau modélisé soit optimisé. L'optimisation étant réalisée en cascade, les biefs amont ne pourront pas anticiper les besoins de biefs plus en aval, ce qui pourrait conduire à des solutions sous-optimales.

Une gestion d'un réseau de voies navigables à l'aide d'une modélisation distribuée de MPC est proposée par [Segovia et al., 2018b] dans la continuation de travaux précédents [Van Overloop et al., 2010, Maestre and Negenborn,]. Cette approche consiste à décomposer la représentation du système en l sous-problèmes et à le résoudre de façon centralisée. Chaque sous-problème possède un ensemble de contraintes définissant sa dynamique et ses interactions avec les autres problèmes. Le problème global sera optimisé itérativement jusqu'à convergence. À chaque itération, la solution de chaque sous-problème sera mise à jour de façon à prendre en compte les choix des autres sous-problèmes. À la fin de l'itération, les paramètres des sous-problèmes sont mis à jour afin de permettre la détection de la convergence. Ainsi une solution optimale au problème original pourra être calculée sur des problèmes en cascade à connaissances complètes.

Contrairement aux solutions précédemment décrites, [Nasir et al., 2016] et [Nasir et al., 2018] propose de prendre en compte des prévisions incertaines sur les déplacements d'eaux non contrôlés. L'approche est utilisée pour optimiser la gestion d'eau d'un seul bief sur un horizon fini. Pour cela une extension stochastique de la commande prédictive est utilisée. Il est ainsi nécessaire d'apprendre un très grand nombre de scénarios d'exécution correspondant aux perturbations possibles. Une solution approximée du problème stochastique pourra ainsi être obtenue. Cependant, le coût de prise en compte des incertitudes ne permet pas *a priori* de passer à l'échelle pour traiter plusieurs biefs.

1.3 Positionnement

Cette thèse vise à optimiser la gestion de la ressource en eau dans les réseaux de voies navigables dans un contexte fortement incertain. Pour cela une modélisation stochastique de l'évolution de ces systèmes est proposée. Le positionnement de ces travaux se fait en comparaisons des approches proposées pour le cadre applicatif des voies navigables.

1.3.1 Récapitulatif des approches existantes

Les approches précédemment décrites se basent sur des hypothèses plus ou moins restrictives afin de fournir des garanties spécifiques à leurs objectifs. Les travaux de cette thèse ont pour but l'obtention d'une solution valide de gestion d'un réseau composé de plusieurs composants indépendants, mais interconnectés ayant des objectifs et des contraintes de gestion spécifiques, tel que le maintien du niveau dans le rectangle de navigation pour chaque bief d'un réseau de voies navigables. Les principaux critères souhaités pour une approche sont la résilience, la prise en compte des incertitudes et la scalabilité. La table 1.1 présente de façon non exhaustive différentes approches d'optimisation de la ressource en eau, à base de satisfaction de contraintes (CSP), d'optimisation quadratique (Quad) et de commande prédictive (MPC). Elles sont illustrées par quelques citations significatives et comparées en fonction des caractéristiques principales concernant la problématique traitée. L'approche défendue dans cette thèse, basée sur les processus décisionnel markovien (MDP), est ajoutée à la fin de la table. Celle-ci et ses différentes caractéristiques seront présentées dans les chapitres suivant.

Caractéristiques de planification

Les méthodologies existantes pour la gestion des voies navigables se différencient par leur capacité de modélisation (réseau, modélisation, gestion et passage à l'échelle), par leur possibilité de rendre la gestion du réseau efficace sur la durée (résilience, incertitudes) et par la qualité des résultats obtenus (optimalité).

Réseau : *cascade* ou *tous*. Les réseaux de voies navigables transfèrent principalement de l'eau de l'amont vers l'aval. Des pompes équipent certains biefs et permettent ainsi des transferts d'eau de l'aval vers l'amont. Plusieurs types de réseau peuvent être définis. Le type de réseau le plus simple est composé de biefs en *cascade*, c'est-à-dire qu'un bief possède au plus un bief en amont et en aval. Dans le cas des voies navigables, un grand nombre de réseau contiennent des bifurcations (diffluent ou confluent) lorsqu'un bief possède plusieurs biefs en amont ou en aval.

Modélisation des biefs : *discrète* ou *continue*. Certaines méthodes d'optimisation ne fonctionnent qu'avec des domaines finis, ce qui rend nécessaire une *discrétisation* des volumes. L'utilisation de domaines *continus* peut rendre l'optimisation plus efficace, car plus adaptée au problème, mais requiert l'accès à des valeurs plus précises de l'état du système.

Résilience : *aucune*, *partielle* ou *complète*. La résilience d'un système est liée à sa capacité à anticiper des événements prévisibles et à récupérer après un imprévu (section 1.1.2, page 11). Une approche ayant une résilience *partielle* possédera soit la capacité d'anticipation soit celle de récupération. Une approche sera *complètement* résiliente si elle peut anticiper et récupérer. La capacité d'anticipation est définie en fonction des connaissances du modèle. Un modèle ne peut anticiper que des événements dont il a une connaissance *a priori*.

Méthode		Réseau	Modélisation	Résilience	Incertitudes	Optimalité	Gestion	Scalabilité	Citation
CSP	Maintien des conditions	tous	discrète	N/A	non	non	volume	moyenne	Nouasse2015
Quad	Non prédictive	tous	continue	aucune	non	non	volume	moyenne	Duviella2016
	Prédictive	tous	continue	partielle	non	oui	volume	moyenne	Duviella2018
MPC	Grand réseau et sous-optimale	cascade	continue	partielle	non	non	débit	grande	Şahin2010
	Déterministe et optimale	cascade	continue	complète	non	oui	débit	moyenne	Segovia2018
	Stochastique	1 bief	continue	complète	oui	non	débit	faible	Nasir2016
MDP	Planification distribuée	tous	discrète	complète	oui	non	volume	grande	Desquesnes2018

TABLE 1.1 – Classification des approches existantes de gestion de l'eau pour des voies navigables

Incertitudes : *oui* ou *non*. Une approche propose une gestion du réseau sous incertitudes si elle prend en compte des événements incertains de façon à les anticiper.

Optimalité : *oui* ou *non*. Une approche est considérée comme optimale, si la planification fournit la meilleure gestion pour le modèle spécifié sur l'horizon considéré. Certaines approches choisissent de se limiter à une solution sous-optimale pour augmenter le passage à l'échelle ou simplement réduire les temps de calcul.

Gestion : *débit* ou *volume*. Ces types de gestions ont été définis dans [Duviella, 2005]. Une gestion de l'eau par *débit* consiste en une régulation en terme de niveau afin de minimiser les impacts des échanges sur le niveau global de l'ouvrage concerné. Ce type de régulation est appliqué sur des problèmes avec une échelle temporelle fine : de l'ordre de l'heure voire des minutes. Les gestions par *volume* visent à répartir et équilibrer des quantités d'eau importante entre plusieurs ouvrages hydrauliques. Ce type de régulation est appliqué davantage sur des problèmes avec une échelle temporelle large : de l'ordre des journées voire des demi-journées ; en considérant un grand nombre d'ouvrages.

Scalabilité : *faible*, *moyenne* ou *grande*. La scalabilité est la capacité d'une approche à passer à l'échelle, c'est-à-dire à représenter des problèmes de tailles réelles. Dans ce tableau, la capacité de passage à l'échelle a été définie pour regrouper les approches pouvant traiter des problèmes de tailles similaires. Les approches à scalabilité *faible* possèdent des difficultés à traiter des problèmes de plusieurs biefs. À l'inverse, les méthodes à *grande* scalabilité pourront gérer des problèmes composés *a priori* sans limites sur le nombre de biefs. Une scalabilité *moyenne* se trouve entre les deux.

Discussion

L'approche par CSP [Nouasse et al., 2015], permet de trouver une solution de gestion pour un réseau de voies navigables sur un seul pas de temps. L'absence de prise en compte d'un horizon de gestion ne permet pas de parler de résilience puisque les événements affectant le réseau sont immédiats et le futur n'est pas pris en compte. L'incapacité de prendre en compte les incertitudes limite la capacité à trouver une solution de gestion valide dans des conditions réelles.

Les deux méthodes basées sur de l'optimisation quadratique [Duviella et al., 2016, Duviella et al., 2018] planifient une gestion sur un horizon de calcul supérieur à 1. Cela rend possible l'anticipation de l'évolution du réseau au cours du temps. Cependant, l'approche pas-à-pas n'utilise pas cet horizon de façon à rendre le réseau partiellement résilient, mais pour obtenir une gestion approchée en un temps de calcul rapide. La seconde version planifie sur un horizon complet ce qui lui confère la capacité d'anticiper et ainsi d'obtenir une meilleure solution. Ces méthodes sont cependant limitées à des scénarios déterministes. Bien que ces deux approches aient une scalabilité moyenne, une approche pas à pas pourra traiter des problèmes de taille plus importante grâce aux optimisations successives de chaque pas de temps.

Grâce à la façon dont elles sont le plus souvent appliquées, les approches basées sur les MPC garantissent au moins une résilience partielle pour chaque modèle. Néanmoins, dans le cas d'une résolution en cascade [Şahin and Morari, 2010], la capacité d'anticipation est limitée à chaque bief seul et non au réseau entier. La méthodologie utilisée dans [Segovia et al., 2018b] permet la résilience d'un réseau composé d'un nombre limité de biefs. Elle ne peut pas *a priori* modéliser tous les types réseaux et n'a pas été appliquées à plusieurs biefs interconnectés. À l'inverse des autres méthodes considérées, [Nasir et al., 2016] propose la prise en compte des incertitudes sur l'horizon de planification considéré afin d'anticiper au maximum ces événements incertains. Cependant, la complexification du modèle ne permet que de représenter un simple bief et non pas un réseau composé de plusieurs biefs.

Ainsi, aucune des méthodes existantes présentées dans la table 1.1 ne correspond idéalement aux critères définis par la problématique de cette thèse ; à savoir modéliser un réseau quelconque pouvant être important pour proposer une gestion résiliente prenant en compte les incertitudes. Bien que préférable, l'obtention d'une solution optimale est secondaire par rapport à la scalabilité dans cette thèse. Elle permet de fournir aux gestionnaires des solutions de gestion et doit être considérée comme un outil d'aide à la décision. De même, un contrôle fin du niveau de l'eau de chaque bief au cours du temps n'est pas une priorité aux vues des objectifs principaux de résilience et de maintien des rectangles de navigation. Une modélisation volumique ne prenant pas en compte l'impact du débit sur le niveau d'eau permet de simplifier la définition du problème.

1.3.2 Vers une approche différente pour des réseaux importants sous incertitudes

Un grand nombre d'inconnues affecte les voies navigables comme la météo, le trafic ou les activités humaines. À titre d'exemple, le bief Cuinchy-Fontinette, situé dans le nord de la France, est fortement affecté par ces inconnues. Il a été recensé plus de 300 points de rejet sur ses 42 kilomètres de longueur. La prise en compte de ces incertitudes implique une complexification de la gestion de la ressource en eau. Le traitement d'un tel problème sur un grand réseau nécessite la proposition d'approches dédiées pour permettre le passage à l'échelle, telles que la distribution de la modélisation et/ou de l'optimisation de la gestion. Outre la facilitation du passage à l'échelle, la distribution du modèle permet aussi une simplification de l'extension du modèle lorsqu'elle est nécessaire. L'ajout de nouveaux éléments sur le réseau, par exemple le canal Seine-Nord, ne nécessiterait que de modifier les quelques modélisations des éléments affectées par cette construction et non l'intégrité du modèle du réseau.

La solution proposée ici s'articule autour de trois contributions. Dans un premier temps, je propose une modélisation générique sous incertitudes permettant de représenter la gestion d'une ressource distribuée sur un réseau afin de pouvoir le rendre résilient. La taille des problèmes rend difficile leur modélisation et leur résolution. De ce fait, la seconde contribution vise à augmenter la capacité à passer à l'échelle en distribuant la modélisation et l'optimisation de la gestion de la ressource. Finalement, une dernière contribution consiste à appliquer la méthodologie proposée aux voies navigables du nord de la France avec une expérimentation sur des sous-réseaux réalistes puis réels.

Modélisation et optimisation générique d'un réseau

À ma connaissance, la majorité des approches visant à optimiser la gestion de ressource d'un réseau sont des approches comme les MPCs, dont le plan est remis en cause à chaque instant pour réduire les effets des incertitudes. Les approches prenant en compte les incertitudes n'étant appliquées que sur des réseaux de tailles plus réduites.

Au vu de la nécessité de gérer les incertitudes, une modélisation basée sur les Processus Décisionnels Markoviens (Markovian Decision Processes - MDP) est proposée. Il existe des approches permettant de traiter des MDPs avec des espaces d'états (voire d'actions) continus. Néanmoins, cette modélisation utilise le plus souvent une discrétisation des états et des capacités de contrôle. La discrétisation des états d'un bief donne l'avantage de pouvoir simplifier la modélisation physique en ignorant par exemple, les effets de vague liés aux transferts d'eau. Cette modélisation permet de planifier pour tous les états possibles du système et en théorie d'avoir une solution optimale même lorsqu'un événement imprévu se produit. Ainsi cette approche peut correspondre aux trois attentes de modélisation d'un réseau quelconque, de prise en compte des incertitudes et de résilience. Ce type d'approche permet d'obtenir des plans de gestion complets sur l'horizon de planification. Ceci permet de les valider *a priori*, mais aussi d'adapter rapidement la gestion

en cas de changement brusque et imprévu de l'état.

La première contribution de cette thèse consiste donc en une modélisation générique par MDP d'un système constitué de plusieurs composants interconnectés, où chaque composant possède une ressource à réguler. Le but étant d'optimiser l'utilisation de la ressource de chaque composant en toute circonstance.

Résolution distribuée

La gestion d'une ressource distribuée dans un réseau définit une classe de problèmes (nommée *Gestion prédictive sous incertitudes d'une ressource partagée sur un réseau*) qui, modélisée avec un MDP, est difficile à résoudre.

En effet, les modélisations par MDP souffrent d'une incapacité à passer à l'échelle. Un grand nombre de sous-classes ont été proposées afin de tirer profit de caractéristiques spécifiques du système modélisé pour augmenter la capacité à passer à l'échelle. Néanmoins, celles-ci ne sont pas adaptées à la classe de problèmes étudiée dans cette thèse.

La deuxième contribution consiste en une distribution de la modélisation sur plusieurs agents, couplée à un algorithme de résolution distribuée grâce à un protocole de communication locale. Cette résolution distribuée limite les échanges d'informations entre deux agents, au strict nécessaire. Limiter les échanges d'informations possède l'avantage d'augmenter la capacité à passer à l'échelle, mais aussi de proposer un certain degré de privacité dans la gestion locale, au détriment de la qualité de la solution obtenue.

La distribution du modèle en agents pose la question du choix de décomposition. Une bonne décomposition permettra de maximiser la scalabilité et la qualité de la solution. Des règles génériques de décomposition seront proposées pour obtenir des décompositions de qualité. Une approche gloutonne de décomposition sera utilisée pour comparer cette approche distribuée à des approches de la littérature compatibles avec la classe de problème précédemment définie.

Applications aux voies navigables

Dans un dernier temps, la méthodologie introduite par cette thèse est appliquée à la gestion des réseaux de voies navigables. Les différents choix de modélisation (discrétisation des volumes et des niveaux d'eau et décomposition en agents) seront formellement présentés. Les politiques de gestion produites seront validées sur diverses simulations d'évolution du réseau. Ceci permettra d'évaluer les solutions sur leur capacité à respecter les contraintes des rectangles de navigation, les écarts des biefs envers leur NNL mais aussi quant à la capacité d'anticiper et de récupérer des événements dégradant le réseau.

Deux types de réseaux de voies navigables des Hauts-de-France sont modélisés. Un réseau de trois biefs (couvrant environ 90 kilomètres) qui permet des comparaisons avec d'autres approches existantes sur trois types de scénarios :

1. déterministe : tous les déplacements d'eau sont préalablement connus ;
2. stochastique connu : les distributions de probabilités et les effets des événements possibles sont connus ;
3. stochastique inconnu : les distributions de probabilités et les effets des événements possibles sont inconnus.

Et un réseau de sept biefs (couvrant environ 110 kilomètres) qui permet de mettre à l'épreuve la capacité de passage à l'échelle de l'approche proposée.

1.4 Conclusion

Ce chapitre a débuté par une description des réseaux de voies navigables et par une présentation de leur fonctionnement actuel. Leur principal objectif de gestion consiste à garantir à chaque instant les conditions de navigation ; i.e. le rectangle de navigation. Il s'agit alors de bien gérer la ressource en eau. Les objectifs de gestion de la ressource en eau vont cependant se trouver perturbés par les évolutions attendues des réseaux dues au changement climatique et à des choix politiques ; tels que l'augmentation de la navigation ou la construction de nouveaux canaux. Ces changements futurs vont augmenter significativement les incertitudes affectant les réseaux. De ce fait, ces évolutions possibles vont rendre plus important le besoin d'avoir une planification sous incertitudes de la gestion de la ressource en eau afin de rendre résilient les réseaux de voies navigable.

Différentes approches de gestion de la ressource en eau ont été décrites. Des méthodes basées sur les problèmes de satisfaction de contraintes, d'optimisation quadratique ou encore de commande prédictive ont été présentées. Leur capacité d'anticiper des événements connus et certains a été discutée. Ces approches peuvent traiter des problèmes de tailles variables avec des solutions acceptables. Cependant, les événements incertains ne peuvent être pris en compte dans des réseaux de tailles importantes. Ainsi les capacités de résilience de la gestion de la ressource en eau sont limitées.

Pour ce faire, à travers les contributions de cette thèse, une approche générique pouvant modéliser l'évolution incertaine des voies navigables est définie. Elle permet de calculer des politiques de gestion couvrant tous les états possibles du système tout en prenant en compte les incertitudes afin de garantir la résilience du réseau. Pour faire passer à l'échelle une telle approche, une distribution de la modélisation et de l'optimisation associée sont proposées.

Chapitre 2

Modélisation par MDP de problèmes de planification sous incertitudes

La planification classique [Fikes and Nilsson, 1971], [McDermott et al., 1998] consiste à trouver une séquence d'actions successives à réaliser afin de résoudre un problème d'optimisation sur un horizon donné. Cette planification suppose une connaissance complète et déterministe du problème d'optimisation traité. Or un grand nombre d'applications réelles sont stochastiques. Par exemple, il n'est pas possible de déterminer de manière sûre la météo ou l'instant et le lieu d'occurrence d'un accident. De ce fait, l'hypothèse d'une connaissance parfaite du problème est très forte et n'est que rarement envisageable pour la mise en œuvre de stratégies de gestion réelles.

La planification stochastique est une catégorie de planification où les résultats des actions sont partiellement aléatoires et partiellement contrôlables. Dans ce cas, le but n'est plus d'obtenir une séquence d'actions à effectuer, mais de déterminer pour chaque configuration possible de systèmes, le meilleur choix possible d'action à effectuer. De nombreuses approches ont été proposées afin de permettre la modélisation de la prise de décision sous incertitudes [Sigaud and Buffet, 2008] [Beynier, 2006] en étendant ou en généralisant les processus décisionnels markoviens [Bellman, 1957].

Ce chapitre présente des outils de modélisation de systèmes permettant une planification sous incertitudes. Dans un premier temps, la formalisation des processus décisionnels markoviens est présentée avec des algorithmes d'optimisation. Un exemple d'application reprenant le problème du « Coffee Robot » [Boutilier et al., 2000] est utilisé pour illustrer ce type de modélisation. Des généralisations aux systèmes multi-agents et aux systèmes partiellement observables sont présentées par la suite. Enfin, un focus spécifique est réalisé sur les méthodes permettant d'outrepasser les limitations de passage à l'échelle de ce type d'approche.

2.1 Processus décisionnel markovien (MDP)

Les processus décisionnels markoviens permettent de modéliser l'évolution d'un système stochastique. Celui-ci peut être contrôlé par des actions et possède des récompenses liées à son évolution. Ainsi, il sera possible de calculer quelles actions doivent être réalisées dans chaque configuration du système pour optimiser sa gestion.

2.1.1 Formalisme

Un processus décisionnel markovien (MDP - Markovian Decision Process) est défini par un tuple $\langle S,A,T,R\rangle$ [Bellman, 1957, Puterman, 1994] qui représente respectivement un espace fini d'états S, un espace fini d'actions A, une fonction de transition T et une fonction de récompense R. Ce modèle définit un système et son évolution selon ses capacités de contrôle. La figure 2.1 donne un exemple de MDP.

Dans cette modélisation, un état correspond à une capture du système dans une configuration particulière. Une action est un événement contrôlé qui peut modifier l'état du système. Les espaces d'états et d'actions représentent respectivement l'ensemble des états possibles dans lesquels le système peut se trouver et l'ensemble des actions possibles pouvant être effectuées sur ce système. La fonction de transition T est définie par $T : S \times A \times S \to [0, 1]$. Elle donne la probabilité $T(s,a,s')$ d'arriver dans un état s' après avoir effectué l'action a dans l'état s. Avoir $T(s,a,s') = 1$ (resp. 0) signifiera qu'atteindre l'état s' en effectuant l'action a depuis l'état s est certain (resp. impossible). Afin d'être cohérent par rapport à l'évolution du système modélisé la somme des probabilités de transition lorsque l'action a est réalisée depuis s est égale à 1 (équation 2.1). C'est-à-dire qu'il est certain d'atteindre au moins un état après avoir effectué une action.

$$\sum_{s' \in S} T(s,a,s') = 1, \forall s,a \in S \times A \tag{2.1}$$

R est la fonction de récompense qui exprime les préférences sur les actions en fonction de l'état de départ et des états atteignables. Elle est définie par $R : S \times A \times S \to \mathbb{R}$, où $R(s,a,s')$ retourne la récompense obtenue si l'état s' est atteint après avoir exécuté l'action a depuis l'état s. Une version $R(s,a)$ est parfois utilisée donnant simplement la valeur de l'exécution d'une action dans un état donné :

$$R(s,a) = \sum_{s' \in S} T(s,a,s') \times R(s,a,s') \tag{2.2}$$

Lorsque toutes les récompenses sont négatives, la fonction est parfois appelée fonction de coût notée $C = -R$.

2.1.2 Politique optimale

Une politique est l'association d'une action à réaliser à partir de chaque état du modèle. [Bellman, 1957] propose une évaluation récursive d'une politique à partir d'un état, voir équation

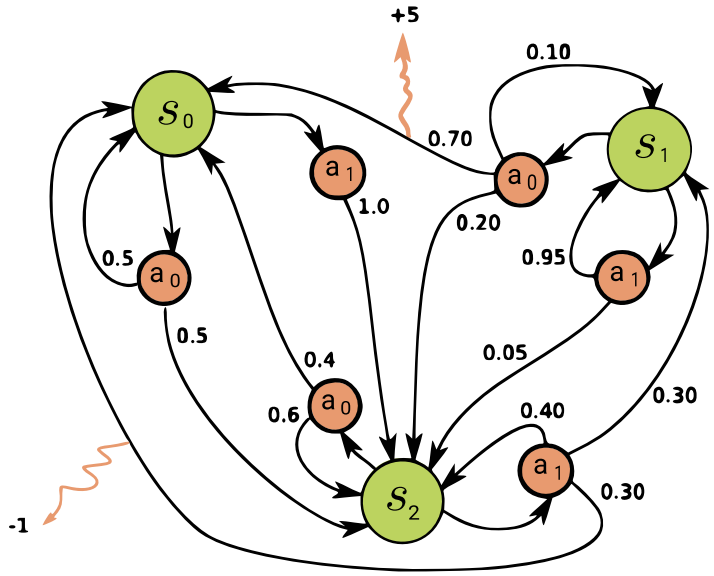

FIGURE 2.1 – Exemple de MDP à trois états (s_0, s_1 et s_2), deux actions (a_0 et a_1). Les transitions non nulles sont représentées par les arcs noirs (e.g. $T(s_1,a_1,s_2) = 0{,}05$) et les récompenses non nulles par les flèches colorées partant des arcs (e.g. $R(s_1,a_0,s_0) = +5$)[4].

2.3. La valeur d'un état est ainsi définie comme la somme espérée des gains immédiats ($R(s,a,s')$) et des gains futurs ($\gamma V^{\pi}(s')$) selon une politique π.

$$V^{\pi}(s) = \sum_{s' \in S} T(s,\pi(s),s') \times (R(s,\pi(s),s') + \gamma V^{\pi}(s')) \tag{2.3}$$

Le paramètre $\gamma \in [0, 1]$ est un facteur d'atténuation. Il est utilisé pour spécifier l'importance des gains futurs par rapport aux gains immédiats.

Une politique optimale est une politique telle qu'aucune autre ne fournit de meilleure évaluation pour un état selon l'équation 2.3, i.e. :

$$V^{\pi^*}(s) \geq V^{\pi}(s) \qquad \forall s \in S \qquad \forall \pi \in A^S \tag{2.4}$$

Formellement π^* est défini par :

$$\pi^*(s) = \arg\max_{a \in A} \left(\sum_{s' \in S} T(s,a,s') \times (R(s,a,s') + \gamma V^{\pi^*}(s)) \right) \qquad \forall s \in S \tag{2.5}$$

4. `https://en.wikipedia.org/wiki/Markov_decision_process`

La recherche d'une solution optimale d'un MDP consiste à trouver une politique $\pi^* : S \rightarrow A$ maximisant les récompenses espérées pour chaque état. Il est possible qu'un modèle possède plusieurs politiques optimales, dans ce cas l'algorithme d'optimisation utilisera des préférences sur les différents choix d'action optimaux. Sur l'exemple de la figure 2.1, la politique optimale du modèle peut se deviner. Une seule action fournit une récompense positive. Le but sera donc de maximiser le nombre de fois où cette action sera réalisée. Ainsi dans l'état s_1 le choix optimal est de faire l'action a_0 qui peut donner une récompense de 5 en atteignant l'état s_0, ou faire un retour en arrière en cas d'échec. L'état s_2 est le seul état permettant d'atteindre s_1 en effectuant l'action a_1. Cette action a aussi une probabilité de conduire à l'état s_0. La pénalité de -1 obtenue lors de l'arrivée dans l'état s_0 est cependant négligeable par rapport aux gains futurs possibles. s_0 ne peut pas atteindre directement s_1. Il faut donc passer par s_2 selon l'action la plus efficace : a_1.

2.1.3 Résolution d'un MDP

Plusieurs approches existent pour calculer une politique optimale d'un MDP. Basé sur l'équation de Bellman (équation 2.3), l'algorithme *Value iteration* [Bellman, 1957] met en œuvre des techniques de programmation dynamique pour calculer, par améliorations successives, la récompense espérée de chaque état (algorithme 1). À partir des récompenses espérées de chaque état ainsi obtenues, il sera possible de déterminer une politique optimale.

Algorithme 1 Value iteration

Données $\langle S,A,T,R\rangle$, γ, ϵ
Résultats π^* et V^{π^*}

1: $V_0(s) \leftarrow 0$, $\forall s \in S$
2: $i \leftarrow 0$
3: **répéter**
4: $\quad i \leftarrow i + 1$
5: $\quad V_i(s) \leftarrow \max_{a \in A} \sum_{s' \in S} T(s,a,s') \times (R(s,a,s') + \gamma V_{i-1}(s))$, $\forall s \in S$
6: **jusqu'à** $|V_i(s) - V_{i-1}(s)| < \epsilon$, $\forall s \in S$
7: **pour chaque** état $s \in S$ **faire**
8: $\quad \pi^*(s) \leftarrow \arg\max_{a \in A} \sum_{s' \in S} T(s,a,s') \times (R(s,a,s') + \gamma V_i(s))$
9: **fin pour**
10: $V^{\pi^*} \leftarrow V_i$

Cet algorithme part d'une estimation de la valeur des états V_0, une bonne estimation initiale pouvant permettre une réduction du temps de calcul et une convergence plus rapide vers une politique optimale. L'algorithme améliore itérativement la qualité de V_i afin de maximiser l'équation de Bellman pour tout état s. Une fois la convergence des estimations de valeurs des

états atteinte, une dernière itération est effectuée pour déterminer pour chaque état quelle action offre la meilleure récompense.

Le facteur d'atténuation γ permet de moduler l'impact des choix futurs et permet donc de résoudre le problème d'optimisation sur un horizon infini. Puisque les gains à l'infini sont atténués progressivement, la valeur ϵ permet de conditionner l'arrêt du calcul. Cette valeur ϵ défini une borne sur l'erreur de la fonction de valeur calculée ($V^{\pi}*$) par rapport à la fonction de valeur optimale ($V*$). Lorsque les augmentations de valeur des états sont considérées comme négligeables, le calcul se termine. Dans le cas d'un horizon fini, il est possible d'utiliser $\gamma = 1$. Cependant dans ce cas, le choix de V_0 pourra avoir un impact important.

Contrairement à l'algorithme *Value Iteration*, l'algorithme *Policy Iteration* [Howard, 1964] manipule directement les politiques. Cet algorithme est initialisé avec une politique. À chaque itération, la fonction de valeur de la politique est calculée selon l'équation 2.3. La politique est mise à jour de façon à maximiser les gains de cette fonction de valeur. L'algorithme s'arrête lorsque la politique n'est plus modifiée. Les itérations de l'algorithme *Policy Iteration* sont plus coûteuses que celle de l'algorithme *Value Iteration* [Puterman, 1994], mais seront moins nombreuses. Ces deux algorithmes conduisent à des résultats similaires et peuvent traiter des problèmes de l'ordre du million d'états [Sutton and Barto, 1998].

Algorithme 2 Policy iteration

Données $\langle S,A,T,R\rangle$, γ

Résultats π^* et V^{π^*}

1: Soit π' une politique du modèle
2: **répéter**
3: $\quad \pi = \pi'$
4: $\quad$ Calculer V^{π} l'évaluation de π
5: $\quad$ Tel que $\forall s \in S$, $V^{\pi}(s) = \sum_{s' \in S} T(s,\pi(s),s') \times (R(s,\pi(s),s') + \gamma V^{\pi}(s'))$
6: $\quad$ **pour chaque** état $s \in S$ **faire**
7: $\quad\quad \pi'(s) \leftarrow \arg\max_{a \in A} \sum_{s' \in S} T(s,a,s') \times (R(s,a,s') + \gamma V^{\pi}(s))$
8: $\quad$ **fin pour**
9: **jusqu'à** $\pi = \pi'$
10: $\pi^* \leftarrow \pi$
11: $V^{\pi^*} \leftarrow V^{\pi}$

2.1.4 Exemple du « Coffee Robot »

Le problème du « Coffee Robot » [Boutilier et al., 2000] est utilisé pour illustrer des modélisations basées sur des MDPs. Dans ce problème, un robot a pour but d'aller chercher un café au bistrot pour son utilisateur, l'utilisateur se trouvant dans son bureau. Si il pleut lors du trajet

entre le bistrot et le bureau, le robot sera mouillé. Le robot devant éviter d'être mouillé, il pourra prendre un parapluie situé dans le bureau avant d'en sortir.

Un état du système sera défini par l'ensemble des caractéristiques du problème :
— H : l'utilisateur veut-il du café ? {oui, non}
— C : le robot a-t-il du café ? {oui, non}
— O : quelle est la position du robot ? {bureau, bistrot}
— W : le robot est-il mouillé ? {oui, non}
— U : le robot a-t-il un parapluie ? {oui, non}
— Ra : pleut-il ? {oui, non}

L'espace d'états est défini par $S = H \times C \times O \times W \times U \times Ra$. Il est donc composé de 2^6 états différents. Ainsi $s = (H : \text{oui}, C : \text{oui}, O : \text{bureau}, W : \text{non}, U : \text{non}, Ra : \text{oui})$ est l'état où l'utilisateur veut du café, le robot est sec et se trouve dans le bureau avec du café et sans parapluie, il pleut.

Afin de réaliser ses objectifs, le robot peut à chaque instant utiliser l'une des quatre actions suivantes :

1. Dep : changer de position (bureau $\leftrightarrow$ bistrot).
2. Obt : obtenir du café ssi (si et seulement si) il est au bistrot.
3. Don : donner le café ssi il a un café et s'il est au bureau.
4. Par : prendre un parapluie ssi il se trouve au bureau.

L'ensemble d'actions est défini par : $A = \{Dep, Obt, Don, Par\}$. Ces actions sont stochastiques. Par exemple, le robot pourrait renverser le café sur le trajet ou lorsqu'il le donne.

Lorsque le robot a servi un café à l'utilisateur, il reçoit une récompense de 0,90. Une faible récompense de 0,10 lui est aussi fournie tant qu'il reste sec. Ces récompenses sont additives. Par exemple, si l'utilisateur a reçu son café et n'en veut plus, et si le robot est sec ($\{H : non, W : non\}$), une récompense de $0{,}90 + 0{,}10$ sera attribuée. La fonction de récompense valorisera ces états. Elle pourra s'écrire de la façon suivante :

$$R(s,a,s') = \begin{cases} 1 & \text{si} \quad s' \cap (H : non, W : non) \neq \emptyset \\ 0{,}90 & \text{si} \quad s' \cap (H : non, W : oui) \neq \emptyset \\ 0{,}10 & \text{si} \quad s' \cap (H : oui, W : non) \neq \emptyset \\ 0 & \text{sinon} \end{cases} \tag{2.6}$$

Dans cet exemple et contrairement à l'exemple de la figure 2.1, les valeurs de la fonction de récompense ne dépendent pas de l'état de départ ni de l'action réalisée. L'impact de l'état de départ et de l'action se trouvera uniquement dans la fonction de transition.

La pluie est un événement aléatoire qui n'est pas affecté par les choix du robot. S'il pleut, la probabilité que la pluie continue est de 0,75 ; inversement, lorsqu'il ne pleut pas il y a une probabilité de 0,30 qu'il commence à pleuvoir à l'instant suivant. Lorsque le robot souhaite livrer un café, il a une probabilité de 0,10 de le renverser. Le fait de renverser un café n'a pas de

pénalité autre que celle de retarder la satisfaction de l'utilisateur. Reprenons l'état $s = (H$: oui, C : oui, O : bureau, W : non, U : non, Ra : oui) défini précédemment ; dans le cas où le robot choisit de donner le café, plusieurs états seront atteignables correspondant aux événements stochastiques possibles :

— $s1 = (\boldsymbol{H}$: **non**, C : non, O : bureau, W : non, U : non, $\boldsymbol{Ra}$: **oui**)
— $s2 = (\boldsymbol{H}$: **non**, C : non, O : bureau, W : non, U : non, $\boldsymbol{Ra}$: **non**)
— $s3 = (\boldsymbol{H}$: **oui**, C : non, O : bureau, W : non, U : non, $\boldsymbol{Ra}$: **oui**)
— $s4 = (\boldsymbol{H}$: **oui**, C : non, O : bureau, W : non, U : non, $\boldsymbol{Ra}$: **non**)

Les états $s1$ et $s2$ correspondent aux états pouvant être atteints lorsque le robot donne le café (H : non), selon l'évolution de la pluie (Ra), et les états $s3$ et $s4$ lorsque le café est renversé (H : oui). Ainsi la fonction de transition de l'état s pour l'action Don peut s'écrire :

$$T(s,Don,s') = \begin{cases} 0{,}675 = 0{,}90 \times 0{,}75 & \text{si} \quad s' = s1 \\ 0{,}225 = 0{,}90 \times 0{,}25 & \text{si} \quad s' = s2 \\ 0{,}075 = 0{,}10 \times 0{,}75 & \text{si} \quad s' = s3 \\ 0{,}025 = 0{,}10 \times 0{,}25 & \text{si} \quad s' = s4 \\ 0{,}0 & \text{sinon} \end{cases} \tag{2.7}$$

Une représentation graphique de la fonction de transition, limitée à l'action Don et aux états s, s_1, s_2, s_3 et s_4, est visible sur la figure 2.2.

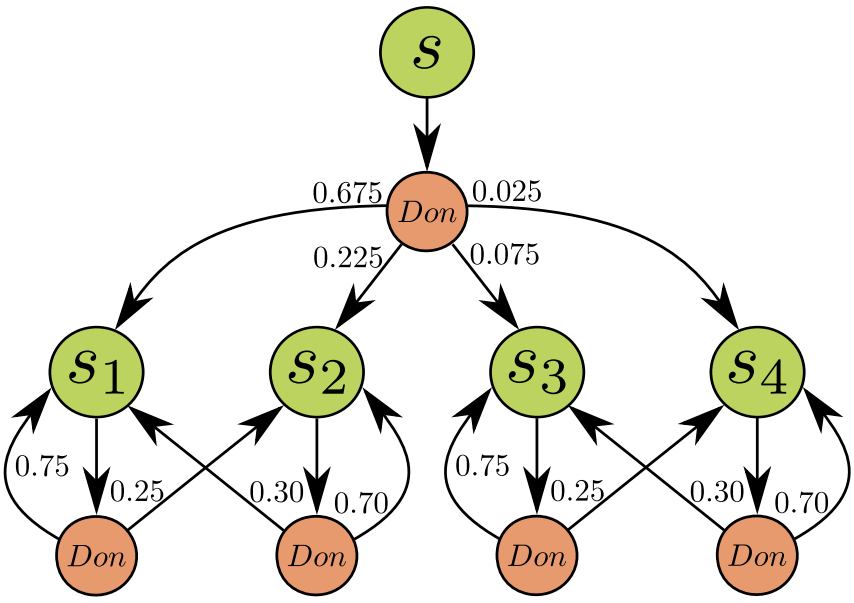

FIGURE 2.2 – Représentation graphique de la fonction de transition pour l'action Don, limitée aux états s, s_1, s_2, s_3 et s_4

Cet exemple a déjà $2^6 = 64$ états et 4 actions. Sa fonction de transition devra représenter

$|S| \times |A| \times |S| = 16384$ combinaisons. Si le nombre de descripteurs est augmenté, par exemple par l'ajout d'un statut pour les portes du bureau et du bistrot (ouverte ou fermée) ou un niveau de charge au robot, le nombre d'états et par corrélation la taille de la fonction de transition augmenteront de façon exponentielle.

Plusieurs robots peuvent évoluer dans un même environnement, chacun avec son ensemble d'actions possibles et potentiellement des objectifs différents. Dans ce cas, il est possible de considérer des approches multi-agents.

2.2 Agentification des MDPs

Les MDPs sont une partie de l'intelligence artificielle qui permet de rendre intelligent une entité en lui donnant la capacité d'agir pour réaliser ses objectifs. Ces entités sont le plus souvent appelées des agents.

2.2.1 Définition d'un agent

Plusieurs définitions d'agents ont été proposées dans la littérature, sans qu'aucune ne soit universellement acceptée. Une définition simple d'un agent a été proposée par [Russell and Norvig, 2003] : un agent est simplement quelque chose qui agit. Des définitions plus complexes ont été proposées afin de définir plus précisément les capacités et les caractéristiques d'un agent. [Franklin and Graesser, 1996] en dresse une liste non exhaustive. Une définition plus précise d'agent proposée par [Wooldridge and Jennings, 1995] est : un agent est un système informatique situé dans un environnement qui est capable d'actions autonomes dans cet environnement de façon à atteindre ses objectifs.

Cette définition introduit la notion d'environnement dans lequel évolue un agent et d'autonomie. Un agent pourra observer son environnement grâce à l'ensemble des capteurs mis à sa disposition et interagir dedans ou le modifier grâce à ses effecteurs (parfois appelés actionneurs). Dans cette thèse, l'autonomie d'un agent est considérée en fonction des apports externes. En dehors d'une description initiale du problème et des objectifs, l'agent ne recevra pas d'information supplémentaire d'agents (humain ou non) avec lesquels il n'est pas en interaction.

La définition d'agent utilisée pour cette thèse se base sur la définition précédente : un agent est une entité située dans un environnement qui est capable d'observer et d'interagir de façon autonome avec cet environnement pour atteindre les objectifs qui lui ont été attribués. Un agent peut communiquer avec d'autres agents pour y parvenir. Enfin, le résultat de ses actions peut être incertain.

Contrairement à la définition d'agent utilisée [Wooldridge and Jennings, 1995], aucune contrainte n'est placée sur le type d'agent. Par exemple, nous considérons qu'un gestionnaire des voies navigables peut être considéré comme un agent qui interagit avec les voies navigables (son environnement) qu'il observe via des capteurs de niveau et agit avec les barrages et vannes pour

maintenir un niveau (l'objectif). Il a la possibilité de communiquer avec d'autres gestionnaires pour réaliser ses objectifs afin par exemple de se coordonner.

Au vu de cette définition, un agent peut être défini par un processus décisionnel markovien. Ceci suppose que l'environnement de l'agent respecte la propriété de Markov, c'est-à-dire que la connaissance du passé n'est pas nécessaire pour déterminer son évolution. Les états représentent la connaissance de l'agent de l'environnement dans lequel il évolue. Les actions correspondent à celles que l'agent peut effectuer. La fonction de transition reflète les évolutions possibles de l'environnement selon les choix de l'agent. La fonction de récompense correspondra aux objectifs ou aux préférences de l'agent.

Dans le cas d'un système multi-agent, les agents évoluent dans un même environnement chacun avec ses objectifs propres. Les objectifs de deux agents peuvent être contradictoires ou se renforcer. De plus, les actions d'un agent peuvent avoir des effets sur les états d'un autre agent. Lorsqu'elles sont possibles, des communications entre les agents peuvent permettre d'optimiser les décisions de chaque agent.

2.2.2 Processus décisionnel markovien multi-agent

Un processus décisionnel markovien multi-agent ou MMDP (Multi-agent Markovian Decision Process) [Boutilier, 1996] est une spécialisation des MDPs pour plusieurs agents. Dans un MMDP, chaque agent dispose d'un ensemble d'actions pour résoudre un ensemble de tâches. Le problème global sera considéré comme coopératif. C'est-à-dire qu'il est préférable pour un agent d'aider un autre agent à atteindre ses objectifs plutôt que de ne rien faire.

Un MMDP est défini par un tuple $\langle Ag,S,A,T,R\rangle$. Ag est un ensemble de n agents $\{Ag_1,\ldots,Ag_n\}$. Similairement aux MDPs, S correspond à l'ensemble des états du système, T la fonction de transition reflétant l'évolution du système en fonction des actions effectuées et R la fonction de récompense modélisant les préférences et les objectifs. L'ensemble des actions A est défini comme la combinaison des actions que chaque agent peut effectuer :

$$A = \prod_{i=1}^{n} A_i \tag{2.8}$$

où A_i est l'ensemble des actions possibles du i$^{\text{ème}}$ agent : Ag_i. Une combinaison d'actions de plusieurs agents, $a = (a_1,\ldots,a_n) \in A$, est appelée une action jointe. Contrairement aux actions, les états et les récompenses sont partagés par les agents. Lors de la résolution du MMDP, une politique $\pi_i : S \rightarrow A_i$ sera produite pour chaque agent. La combinaison des politiques $\pi = (\pi_1,\ldots,\pi_n)$ est appelée politique jointe. La modélisation des états et de la fonction de transition étant la même que pour les MDPs, les MMDPs souffrent des mêmes problèmes d'explosion combinatoire que les MDPs.

L'exemple du *Coffee Robot* pourrait être étendu à plusieurs agents, en considérant un agent préparateur (robotisé) possédant deux actions : $A_1 = \{Don,Net\}$. L'action Don est utilisée par

ce nouvel agent pour donner du café au robot livreur si celui-ci se trouve dans la pièce et en fait la demande. Si l'agent préparateur effectue l'action Don, mais que le robot livreur n'est pas en demande de café, le café sera gaspillé, résultant en un coût de 0,10. L'action Net est utilisée par l'agent pour nettoyer le bistrot. Cette action rapporte toujours un gain de 0,05. Les actions du robot livreur de café sont toujours définies par l'ensemble $A_2 = \{Dep,Obt,Don,Par\}$. Cet exemple conserve le même ensemble d'états que le problème original, mais possède un ensemble d'actions plus étendu : $A = A_1 \times A_2$. Les fonctions de transition et de récompense seront modifiées pour prendre en compte le nouvel ensemble d'actions.

Dans le cas où le bistrot serait rempli de clients, il pourrait être difficile au préparateur de déterminer si le livreur est présent dans le bistrot. Pour cela, des généralisations des MDPs prenant en compte qu'il peut ne pas être possible de déterminer avec certitude l'état courant du système ont été proposées.

2.2.3 Processus décisionnel markovien partiellement observable

Les processus décisionnels markoviens partiellement observables (POMDP - Partially Observable Markovian Decision Process) sont une généralisation des MDPs qui a été introduite pour modéliser les problèmes où il n'est pas possible de déterminer avec certitude l'état actuel de l'agent dans le système [Smallwood and Sondik, 1973] [Cassandra et al., 1997]. Celui-ci n'a accès qu'à des observations plus ou moins limitées qu'il utilisera pour inférer les états possibles dans lesquels il peut se trouver.

Un POMDP est défini par un tuple $\langle S,A,T,R,\Omega,O\rangle$. S et A sont respectivement l'ensemble d'états et d'actions, T est la fonction de transition et R la fonction de récompenses. Ceci est comparable à la définition des MDPs. L'ensemble des observations possibles de l'agent est représenté par Ω. Les observations d'un agent ne sont pas forcément précises ; les capteurs pouvant être imprécis, bruités ou les relevés ambigus. Par exemple, un robot peut observer qu'il est dans un couloir, car il observe deux murs parallèles proches, mais ne sait pas forcément dans quel couloir il se trouve. Finalement, $O : S \times A \times S \times \Omega \rightarrow [0, 1]$ est une fonction $O(s,a,s',\omega)$ qui donne la probabilité d'observer ω lorsque l'agent arrive dans l'état s' après avoir effectué l'action a dans l'état s. L'état actuel n'étant pas connu, la politique sera définie sur un historique de k observations ($\pi : \Omega^k \rightarrow A$).

La résolution optimale, à horizon fini, d'un POMDP est plus complexe que celle des MDPs passant de P-Complet (MDP) à NP-Complet (POMDP) [Papadimitriou and Tsitsiklis, 1987]. Les POMDPs à horizon infini sont indécidables [Madani et al., 1999], c'est-à-dire que les solutions optimales ne sont pas atteignables en un nombre fini d'étapes. Des approches heuristiques et approximées ont été définies pour résoudre des POMDPs à horizon infini. Celles-ci se basent sur des états de croyance B [Kaelbling et al., 1998]. Un état de croyance $b_t \in B$ est une distribution de probabilités d'être dans chacun des états $s \in S$ à un instant t.

Dans le cas de l'exemple du *Coffee Robot*, les capteurs du robot peuvent avoir du mal à

détecter si l'humain souhaite avoir ou non du café, notamment lorsque le robot quitte le bureau. Afin d'avoir la planification la plus efficace, le robot devra prendre en compte que ses observations peuvent se dégrader en fonction de son éloignement de l'humain.

Comme les MDPs, les POMDPS ont été étendus afin de permettre de modéliser des problèmes multi-agents.

2.2.4 Dec-POMDP et Dec-MDP

Les processus décisionnels markoviens partiellement observables décentralisés (Dec-POMDP - Decentralised Partially Observable Markovian Decision Process) sont une généralisation des POMDPs permettant de distribuer les capacités de contrôle d'un système sur plusieurs agents [Bernstein et al., 2002] [Seuken and Zilberstein, 2012]. Chaque agent n'a qu'une observation partielle de l'état global du système.

Un Dec-POMDP est défini par un tuple $\langle Ag,S,A,T,R,\Omega,O\rangle$. Ag est un ensemble de n agents $\{Ag_1,\ldots,Ag_n\}$, S et A sont respectivement les espaces d'états et d'actions jointes du système modélisé. L'ensemble d'actions jointes est défini similairement aux MMDPs comme les produits de l'ensemble d'actions de chaque agent. $T : S \times A \times S \rightarrow [0,1]$ et $R : S \times A \times S \rightarrow \mathbb{R}$ sont respectivement les fonctions de transition et de récompense. Ω est l'ensemble des observations jointes. Cet ensemble est défini comme le produit des observations de chaque agent :

$$\Omega = \prod_{i=1}^{n} \Omega_i \tag{2.9}$$

Ω_i est l'ensemble des observations de l'agent i. Cet ensemble correspond aux valeurs que l'agent peut observer grâce à ses capteurs. La fonction d'observation est définie par

$$O : S \times A \times S \times \Omega \rightarrow [0,1] \tag{2.10}$$

$O(s,a,s',\omega)$ définit la probabilité d'avoir l'observation jointe $\omega = (\omega_1,\ldots,\omega_n)$ si l'état s' est atteint après que les agents aient effectué l'action jointe $a = (a_1,\ldots,a_n)$ depuis l'état s.

Le but est de trouver une politique optimale jointe $\Pi^* = (\pi_1^*,\ldots,\pi_n^*)$ de façon à ce que chaque agent i possède sa propre politique et que la combinaison de ces politiques soit optimale. Pour les Dec-POMDPs, la résolution est centralisée, mais la solution est distribuée sur les agents. Lors de l'exécution, un agent choisira l'action qu'il devra effectuer en fonction de son historique d'observation ou de son état de croyance. Lorsque les agents ont des capacités de communication, celle-ci seront incluses dans le modèle et leur utilisation sera prédéterminée lors de la planification. L'obtention d'une solution optimale d'un Dec-POMDP est encore plus difficile que les POMDPs, passant de NP-Complet à NEXP-Complet, ce qui rend, en pratique, impossible une résolution générique de Dec-POMDP sur des problèmes réalistes.

5. `rbr.cs.umass.edu/camato/decpomdp/overview.html`

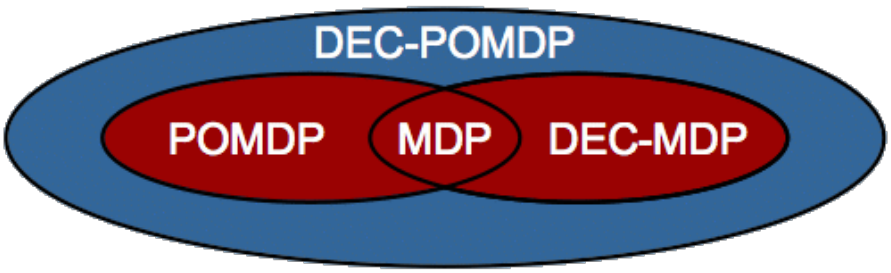

FIGURE 2.3 – Relation en MDP, POMDP, Dec-MDP et POMDP[5]

Enfin, un processus décisionnel markovien décentralisé (Dec-MDP) est un Dec-POMDP où l'état du système est collectivement pleinement observable [Bernstein et al., 2002]. C'est-à-dire qu'à partir des observations de tous les agents, il est possible de déterminer l'état exact du système. Pour un Dec-MDP, il est possible de déduire l'état grâce à l'observation jointe suite à une action, donc $O : S \times A \times S \times \Omega \rightarrow \{0, 1\}$. Si chaque agent peut observer pleinement l'état courant du système, le problème sera un MMDP. La relation entre entre MDP, POMDP, Dec-MDP et Dec-POMDP est mise en avant sur la figure 2.3.

Dans l'exemple du *Coffee Robot*, un Dec-POMDP peut être utilisé pour modéliser un problème où plusieurs robots doivent chercher du café pour leur humain respectif. Chaque robot peut observer le souhait de son humain d'avoir du café. Lorsque plusieurs robots demandent un café simultanément dans le bistrot, seul un l'obtiendra. Cependant, les robots n'ont qu'une observation imprécise de ce qu'il transporte et ne peuvent pas déterminer s'ils ont obtenu un café. Le Dec-POMDP fournira à chaque robot une politique permettant d'éviter ces conflits afin de maximiser la satisfaction des humains.

2.3 Modélisations par MDP centralisée pour passer à l'échelle

Les MDPs sont de puissants outils de modélisation. Cependant, et sans même considérer les contraintes d'observation limitée, ils souffrent d'un problème important de passage à l'échelle. La taille des modèles étant exponentielle avec le nombre d'états et d'actions, il devient rapidement difficile de les utiliser pour les appliquer à des problèmes complexes et réalistes. Pour cela, diverses méthodes exploitant les propriétés des systèmes à modéliser ont été proposées.

2.3.1 MDP factorisé

Une solution pour faciliter le passage à l'échelle des MDPs consiste à utiliser une représentation compacte des fonctions de transition et de récompense plutôt que d'énumérer tous les

triplets (état, action, état). C'est le cas des approches par MDP factorisé (F-MDP - Factored MDP) [Boutilier et al., 1995, Boutilier et al., 2000, Hoey et al., 1999, Radoszycki et al., 2015]. Une présentation plus complète des MDPs factorisés est proposée dans [Sigaud and Buffet, 2008, chapitre 4].

Les F-MDPs sont applicables aux problèmes dont les états peuvent être divisés en un nombre fini de variables dont les évolutions ne dépendent que du passé. Formellement, pour un F-MDP les états sont définis par une variable aléatoire multivariée, aussi appelée vecteur aléatoire :

$$X = [X_1, \ldots, X_m] \tag{2.11}$$

où $Dom(X_i)$ est le domaine de la variable X_i. $Dom(X)$ représente l'ensemble des assignations possibles du vecteur X et donc : $Dom(X) = S$. Un état du domaine est donc une assignation possible de X :

$$s = (x_1, \ldots, x_m) \text{ avec } x_i \in Dom(X_i) \tag{2.12}$$

Comme les états ne sont pas atomiques, c'est-à-dire qu'il est possible de les diviser en une combinaison de variables, il est possible d'exploiter l'indépendance de l'évolution de leurs variables pour représenter de façon compacte les fonctions de transition et de récompense. Pour chaque variable X_i, l'ensemble $Parent_a(X_i) \subseteq X$ définit les variables dont la valeur à l'instant t influencera la valeur de X_i à l'instant $t+1$ lorsque l'action a sera appliquée. Par définition, les évolutions des X_i sont indépendantes les unes des autres, alors il est possible de réécrire la fonction de transition sous la forme suivante :

$$T(s,a,s') = \prod_{i=1}^{m} T(s,a,x'_i) = \prod_{i=1}^{m} P(x'_i|a,(Parent_a(x'_i)) \tag{2.13}$$

Similairement à la fonction de transition, la fonction de récompense peut être décomposée en une somme de fonctions de récompense locale définies sur un sous-ensemble de variables. Un exemple est donné ci-après.

Différentes représentations factorisées des fonctions de transition et de récompense ont été proposées. Par exemple, les diagrammes de décision algébrique sont des graphes acycliques composés de nœuds terminaux représentant les valeurs possibles de la fonction de transition. Chaque nœud non terminal représente une variable et les arcs en sortant ses valuations possibles. Leur réduction permet une représentation compacte des fonctions de transition et de récompense [Hoey et al., 1999]. Des règles sont aussi utilisées pour les modélisations [Guestrin et al., 2003, Guestrin and Koller, 2003]. Une règle est composée d'une contrainte sur la valuation des variables présentes et futures et d'une valeur obtenue si cette contrainte est respectée. Les algorithmes de recherche de politique optimale ont ainsi été adaptés à ces variantes de modélisation.

L'exemple du *Coffee Robot* est parfaitement adapté pour être modélisé par un F-MDP ; celui-ci ayant été défini pour illustrer ce concept. L'état est décrit par les 6 variables booléennes définies précédemment :

O^t	U^t	$U^{t+1} = oui$
non	non	0,0
non	oui	1,0
oui	non	0,9
oui	oui	1,0

TABLE 2.1 – $P(U^{t+1}|Par,Parents(U))$: probabilité d'évolution de la variable U lors de l'action déterministe Par

— H : l'utilisateur veut-il du café ?
— C : le robot a-t-il un café ?
— O : le robot est-il au bureau ? (sinon il sera au bistrot)
— W : le robot est-il mouillé ?
— U : le robot a-t-il un parapluie ?
— Ra : pleut-il ?

Ainsi que par les quatre actions :

1. Dep : changer de position (bureau $\leftrightarrow$ bistrot).
2. Obt : obtenir du café ssi il est au bistrot.
3. Don : donner le café ssi il a un café et s'il est au bureau.
4. Par : prendre un parapluie ssi il se trouve au bureau.

Un état $s = (H:non, C:oui, O:oui, W:oui, U:non, Ra:oui)$ est l'assignation d'une valeur à chaque variable. Or l'évolution de chaque variable, selon les actions effectuées, ne dépend que d'un faible nombre de variables. Prenons comme exemple, la variable U correspondant au port d'un parapluie par le robot. Lorsque l'action de prise de parapluie, Par, est effectuée à l'instant t, le port du parapluie à l'instant suivant, U^{t+1}, ne dépendra que de la position du robot O^t et de s'il portait ou non un parapluie U^t. Ceci permet de représenter de façon compacte la probabilité d'évolution de la variable U en fonction de l'action Par, voir table 2.1. Cette table rend visible que si le robot possède déjà un parapluie ou si il ne se trouve pas au bureau son état ne changera pas ; autrement il aura une probabilité de 0,9 de ramasser un parapluie. Les autres variables n'étant pas affectées par l'action, leur évaluation au temps $t+1$ ne dépendra que de leur valeur au temps t, à l'exception de Ra. La pluie n'est pas une variable dont la valeur est contrôlée par le robot. Elle évolue stochastiquement par rapport à sa valuation précédente.

Il est ainsi possible d'utiliser l'équation 2.13 pour écrire la fonction de transition pour cette action :

$$T(s,Par,s') = P(u'|Par,u,o) \times \prod_{x \in \{h,c,o,w,ra\}} P(x'|Par,x) \tag{2.14}$$

La valuation des variables de l'état s (et similairement pour s') est représentée par une notation en minuscule : h, c, o, w, u et ra.

Cette écriture synthétique de la fonction de transition pour l'action Par nécessite de lister seulement $2 \times 2^2 + 5 \times (2 \times 2^1) = 28$ assignations possibles des variables. Ici 2^2 aux assignations possibles des parents de la variable u' ; il est multiplié par 2 car u est une variable binaire. Chaque 2^1 correspond aux variables non affectées par l'action Par et qui n'ont donc qu'un seul parent. Par comparaison, un MDP où la fonction de transition prend en compte toutes les possibilités d'états de départ et d'arrivée pour l'action Par listerait $2^6 \times 2^6 = 4096$ assignation possibles.

De façon analogue, la fonction de récompense peut être fortement simplifiée. Celle-ci récompense le robot lorsqu'il est sec, $W = non$, et lorsque l'humain ne veut pas de café $H = non$. Ce qui permet de l'écrire simplement sous la forme suivante :

$$R(s,a,s') = \begin{cases} 0{,}9 & \text{si } h' = \text{ non} \\ 0{,}0 & \text{sinon} \end{cases} + \begin{cases} 0{,}1 & \text{si } w' = \text{ non} \\ 0{,}0 & \text{sinon} \end{cases} \tag{2.15}$$

2.3.2 MDP décomposé

La décomposition de MDP est une autre solution pour permettre de traiter des problèmes de grande taille à l'aide de MDP. Le but est de réduire la complexité de calcul en construisant une hiérarchie entre des problèmes locaux et une solution globale. Les problèmes locaux sont ainsi résolus indépendamment les uns des autres puis sont combinés, ou utilisés dans ces calculs, pour obtenir une solution au problème initial. [Boutilier et al., 1999] propose deux catégories de décompositions : les décompositions en série et les décompositions parallèles.

Les décompositions en série utilisent une séparation de l'espace d'état en partitions faiblement connectées. Chaque partition est modélisée par un sous-MDP. [Dean and Lin, 1995] proposent deux solutions à ce genre de problèmes. La première est une méthode itérative qui consiste à successivement résoudre les sous-MDPs afin d'optimiser un certain nombre de paramètres qui permettront d'améliorer la solution globale. Cette approche garantit la convergence vers une solution optimale, mais pas une accélération de la résolution. Les auteurs considèrent néanmoins qu'un faible nombre d'itérations est requis pour obtenir une solution suffisamment proche de l'optimalité. La seconde, hiérarchique, consiste à définir un ensemble de paramètres pour chaque sous-MDP afin de calculer un ensemble fini de sous-politiques. Un MDP haut-niveau est défini sur des méta-états (les sous-MDPs) et des métas-actions (les sous-politiques) afin de calculer une politique globale cohérente pour le problème initial. Cette méthode permet de contrôler le compromis entre les ressources de calcul (espace mémoire et temps de calcul) de la politique globale, mais ne garantit pas son optimalité.

Les décompositions parallèles divisent le modèle initial en sous-MDPs, où chaque action affecte localement chacun de ses sous-MDPs. Le MDP initial correspond donc à un produit de MDPs locaux. Chaque sous-problème est optimisé localement, puis les résultats obtenus sont fusionnés en une solution globale optimale [Singh and Cohn, 1998] ou une solution globale approximée [Meuleau et al., 1998]. Cependant, ces approches nécessitent que les problèmes locaux

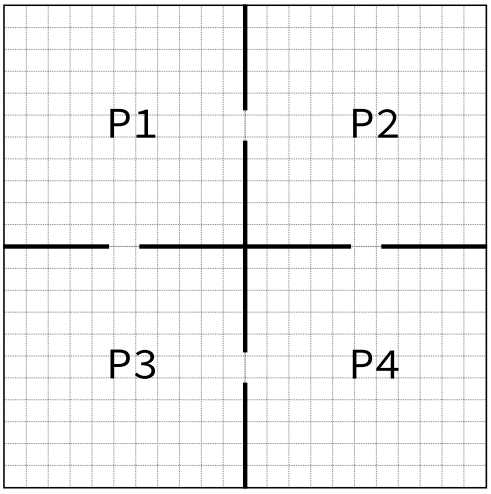

FIGURE 2.4 – Exemple de bâtiments dont les 4 pièces sont faiblement connectées

soient quasiment indépendants, où les relations entre les problèmes sont principalement sur les actions possibles et sur les récompenses. Une méthode automatique de décomposition des problèmes compatibles a été proposée dans [Boutilier et al., 1997].

Les approches décomposées ont la possibilité de réduire la complexité (spatiale et temporelle) de calcul pour un grand nombre d'applications, avec ou sans garantie d'optimalité. Elles nécessitent de calculer un partitionnement de l'espace d'états [Parr, 1998,Sabbadin, 2002]. Cependant, s'il n'y a pas de décomposition évidente, le partitionnement est un problème difficile [Bichot and Siarry, 2011].

Une application possible des MDPs hiérarchiques est un problème où un agent doit effectuer des tâches de nettoyage dans des pièces connectées les unes aux autres par des portes (figure 2.4).

Chaque pièce est modélisée indépendamment des autres par un MDP, contenant les états de la pièce et les actions possibles de l'agent dans cette pièce. Pour chaque pièce, plusieurs politiques seront calculées représentant les actions globales qui peuvent y être effectuées. Par exemple, pour la pièce P1, trois actions globales peuvent être appliquées : aller dans la pièce P2, aller dans la pièce P3 ou nettoyer la pièce actuelle. Des politiques locales à la pièce P1 seront calculées grâce à des heuristiques afin de maximiser les objectifs de ces actions globales.

Le modèle hiérarchique aura donc pour ensemble d'états les MDPs locaux de chaque pièce et pour ensemble des actions toutes les politiques locales. Les fonctions de transition et de récompense de ce modèle seront construites sur ces nouveaux ensembles d'états et d'actions, en fonction des résultats possibles d'applications des politiques locales.

2.3.3 Approche basée Monte-Carlo

Une approche différente pour résoudre les MDPs est d'approximer la politique optimale grâce à l'utilisation d'un simulateur du problème considéré [Kearns et al., 2002, Kocsis and Szepesvári, 2006]. Ce simulateur reçoit en entrée un couple (s,a) et retourne un couple (s',r) correspondant à un état aléatoire, atteignable par ce couple état-action, ainsi que la récompense obtenue. Ce type d'approche consiste à trouver pour un état donné la meilleure action possible sur une durée limitée. Contrairement aux méthodes précédentes de résolution, elle vise à calculer la solution au fur et à mesure de son utilisation.

Les algorithmes de planification Monte-Carlo construisent un arbre de recherche en partant d'un état donné grâce à une suite de simulations aléatoires, potentiellement contrôlées par une heuristique. La racine de l'arbre correspond à l'état initial, les nœuds aux états explorés et les arcs aux actions. Dans le cas de [Kearns et al., 2002], l'arbre est construit en tirant pour chaque action possible C résultats possibles de simulation. L'arbre est ainsi construit récursivement jusqu'à une profondeur fixée. Puis, les estimations de valeurs de chaque état sont remontées des feuilles vers la racine. Il sera ainsi possible de déterminer l'action qui apporte le meilleur gain. Le nombre de résultats à explorer à chaque niveau peut être dégressif. En effet, lorsqu'un paramètre d'atténuation $\gamma < 1$ est utilisé, l'impact des états futurs devient de plus en plus faible avec la distance.

Une autre solution est l'UCT (Upper Confidence Bound applied to Trees) [Kocsis and Szepesvári, 2006]. Dans ce cas, en partant de la racine, des actions sont choisies successivement grâce à une heuristique et les états atteints à la suite des actions sont le résultat d'une simulation. Des actions sont ainsi choisies jusqu'à atteindre un état terminal (e.g. profondeur de l'arbre, aucune action possible). Les évaluations des états sont ensuite mises à jour en partant de la feuille atteinte pour remonter jusqu'à la racine. L'heuristique de choix des actions de l'UCT vise à explorer au moins une fois chaque action disponible par état puis utilise un compromis entre l'exploration d'états inconnus et les états les plus intéressants [Kocsis et al., 2006].

L'algorithme 3 est une version générique des planifications Monte-Carlo, qui consiste en une suite d'explorations de branches jusqu'à atteindre un certain critère ; par exemple le temps d'exécution (ligne 1-3). À la fin de l'exécution, la meilleure action est sélectionnée (ligne 4). Pour chaque exploration de branche, l'évaluation des états et la simulation des actions seront obtenues par le simulateur du modèle traité (ligne 7 et 10). Le choix des actions (ligne 9) sera la principale différence entre les différentes approches. Pour [Kearns et al., 2002], toutes les actions seront simulées un même nombre de fois. Tandis que dans [Kocsis and Szepesvári, 2006], les actions qui semblent être les plus intéressantes seront plus visitées. La mise à jour de l'arbre en fonction des explorations (ligne 12) sera dépendante des critères nécessaires au fonctionnement de la sélection.

Ces solutions permettent un passage à l'échelle de la résolution en ne nécessitant plus de représenter les fonctions de transition dans leur ensemble. En effet, les tirages aléatoires du simulateur respectent la distribution de probabilités correspondant à ces fonctions. Néanmoins

Algorithme 3 Algorithme générique de planification Monte-Carlo

```
répéter
    recherche(état, 0)
jusqu'à arrêt
retourner meilleureAction(état,0)

fonction recherche(état, profondeur)
    si état est une feuille alors
        retourner évaluation(état)
    fin si
    action = choisirAction(état, profondeur)
    (étatSuivant, récompense) = simuler(état, action)
    q = récompense + γ × recherche(étatSuivant, profondeur + 1)
    mettreÀJour(état, action, q, profondeur)
fin fonction
```

puisque l'intégralité de l'espace de recherche n'est pas explorée, l'optimalité des solutions n'est plus garantie.

2.4 Résolution distribuée de MDP

Une autre solution pour résoudre des MDPs consiste à distribuer la modélisation et la résolution sur plusieurs machines. Ces distributions exploitent le plus souvent des caractéristiques du modèle. Les deux grandes catégories de caractéristiques exploitées sont l'indépendance [Nair et al., 2005] et les interactions entre agents [Melo and Veloso, 2013]. Dans le cas du « Coffee Robot », il serait intuitif que chaque robot calcule sa politique à l'aide de ses ressources.

2.4.1 Dec-SIMDP - Modèle à faible interaction

Les processus décisionnels markoviens décentralisés avec de faibles interactions [Melo and Veloso, 2013] (Dec-SIMDP - Decentralised Markovian Decision Process with Sparse Interactions) sont un cas particulier de Dec-MDP où les interactions entre agents sont limitées à un ensemble de zones d'interaction Z de petite taille par rapport au problème original. L'hypothèse est faite que l'état est totalement observable dans les zones d'interactions et qu'il n'y a pas de problèmes de coordination en dehors de ces zones (i.e. l'agent est "seul" avec des objectifs qui lui sont propre). Dans un Dec-SIMDP, les espaces d'états et d'actions sont respectivement les ensembles

d'états joints et d'actions jointes :

$$\begin{aligned} S &= S_1 \times \cdots \times S_n \\ A &= A_1 \times \cdots \times A_n \\ \text{avec } Ag &= \{1,\ldots,n\} \end{aligned} \tag{2.16}$$

De ce fait, il est possible de décomposer la fonction de récompense en une somme de récompenses dépendant uniquement d'un agent i (R_i) et de récompenses liées à la coordination d'un ensemble d'agents K_j sur une zone d'interaction j (R_j^I) :

$$R(s,a,s') = \sum_{i=1}^{n} R_i(s_i,a_i,s'_i) + \sum_{j \in Z} R_j^I(s_{K_j},a_{K_j},s'_{K_j}) \tag{2.17}$$

Cette approche utilise le fait qu'un Dec-MDP peut se réécrire sous la forme d'un tuple :

$$\Gamma = (\{M_i,\ \forall i \in Ag\},\{(S_j^I,M_j^I)\ \forall j \in Z\}) \tag{2.18}$$

où :

— $M_i = \langle S_i,A_i,T_i,R_i \rangle$ est un MDP modélisant le $i^{\text{ème}}$ agent en l'absence d'autre agent. T_i est la fonction de transition propre de cet agent pour les états sans interaction ;

— $M_j^I = \langle K_j, \prod_{k \in K_j} S_k, A_j^I, T_j^I, R_j^I \rangle$ est un MMDP qui modélise les interactions locales d'un sous-ensemble K_j d'agents sur une zone d'interaction $S_j^I \subset \prod_{k \in K_j} S_k$.

Dans le cas où les agents sont indépendants, le Dec-SIMDP sera équivalent à un ensemble de MDPs qui pourront être distribués et optimisés séparément. Dans le cas où l'ensemble des états est composé uniquement d'états d'interactions, le problème sera équivalent à un MMDP, puisque l'hypothèse est faite que les états d'interaction sont pleinement observables.

Une application possible des Dec-SIMDP est la navigation : plusieurs robots se déplacent dans un environnement partagé avec des zones contraintes. Chaque robot a un objectif propre et doit éviter de croiser les autres robots dans les zones contraintes ; par exemple pour franchir un pont étroit, ou une zone glissante. Deux robots présents dans une même zone contrainte reçoivent une pénalité.

L'exemple (figure 2.5) est une adaptation du problème « Coffee Robot ». Pour passer du bistrot au bureau, il est nécessaire de passer par une passerelle. Un seul robot peut emprunter la passerelle à un instant donné. Pour cet exemple, l'espace d'états du problème est augmenté pour prendre en compte de manière plus fine la position des robots dans chaque lieu.

Pour éviter que les robots franchissent la passerelle en même temps et reçoivent ainsi une grosse pénalité, la passerelle et les deux états y amenant directement seront mis en zone d'interaction, en gris sur la figure. En dehors de cette zone, les robots n'ont aucun impact immédiat l'un sur l'autre et peuvent donc s'ignorer. Lorsqu'un robot arrivera dans cette zone, il pourra

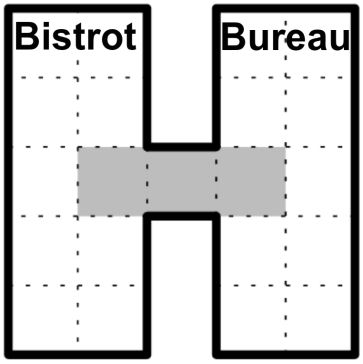

FIGURE 2.5 – Problème du « Coffee Robot »avec des zones d'interactions (en gris)

observer si l'autre robot est présent dans cette zone et agira en conséquence pour maximiser les gains : franchir la zone seul.

Divers algorithmes de planification pour les Dec-SIMDP ont été proposés. Parmi ceux-ci, LAPSI (Look-Ahead Planning for Sparse Interaction) propose une approche intuitive qui fournit de bons résultats. LAPSI repose sur le principe que si tous les agents sauf un ont une observabilité complète de l'environnement et suivent une politique jointe fixée, alors le système se comporte comme un POMDP pour ce dernier. L'algorithme consiste donc pour chaque agent à :

1. faire une hypothèse sur la politique jointe des autres agents, en résolvant le MMDP correspondant ;
2. construire le POMDP associé et le résoudre.

Les Dec-SIMDPs permettent de représenter efficacement des problèmes multi-agents avec un faible nombre de zones d'interaction. Cependant, la taille des MMDPs et des POMDPs a résoudre dépend du nombre d'interactions, ce qui peut rendre difficile le passage à l'échelle d'une telle approche.

2.4.2 ND-POMDP

Un ND-POMDP (Networked Distributed POMDP) [Nair et al., 2005] est une approche distribuée pour la modélisation et la résolution de POMDP multi-agents, où les interactions entre les agents sont très localisées, avec une indépendance des fonctions de transition. L'indépendance des fonctions de transition signifie que les actions d'un agent n'ont pas d'effet sur l'évolution du modèle local des autres agents. Différentes approches ont été proposées pour résoudre les ND-POMDPS de façon optimale (FB-HSVI [Dibangoye et al., 2014]), localement optimale (LID-JESP [Nair et al., 2005]) et approximée (FANS [Marecki et al., 2008]).

Les ND-POMDPs sont une classe particulière de Dec-POMDP. Ils sont définis par un tuple $\langle Ag,S,A,T,R,\Omega,O,b_0\rangle$. Ag est un ensemble de n agents. $S = S_1 \times \cdots \times S_n \times S_u$ est l'ensemble des états du système composé de plusieurs parties. S_i représente l'ensemble des états propres au

$i^{\text{ème}}$ agent et S_u un ensemble d'états qui ne sont affectés par aucune des actions des agents ; par exemple l'écoulement du temps ou la météo. A est un ensemble d'actions jointes. b_0 est un état de croyance sur l'état initial de l'environnement.

Les ND-POMDPs se basent sur l'indépendance des transitions des agents. Cela permet de définir la fonction de transition T du modèle de la façon suivante :

$$T(s,a,s') = T_u(s_u,s'_u) \times \prod_{i=1}^{n} T_i(s_i,s_u,a_i,s'_i) \tag{2.19}$$

avec $a = (a_1,\ldots,a_n)$ l'action jointe effectuée dans l'état $s = (s_1,\ldots,s_n,s_u)$ et $s' = (s'_1,\ldots,s'_n,s'_u)$ l'état atteint. La fonction de transition locale T_i correspond à la probabilité d'atteindre l'état local s'_i lorsque le $i^{\text{ème}}$ agent se trouve dans l'état (s_i,s_u) et effectue l'action a_i. T_u est la fonction de transition des états ne dépendant pas des agents qui donne la probabilité d'atteindre l'état s'_u en partant de s_u.

La fonction de récompense est définie comme une somme de récompenses spécialisées :

$$R = \sum_{l} R((s_{l1},\ldots,s_{lk},s_u),(a_{l1},\ldots a_{lk}),(s'_{l1},\ldots,s'_{lk},s'_u)) \tag{2.20}$$

où chaque l représente un sous-ensemble de k agents obtenant des récompenses en travaillant ensemble. Par exemple, deux caméras de surveillance qui observent une zone commune en même temps sont récompensées, car cela réduit les chances de faux positifs. Deux agents appartenant à un même ensemble l sont définis comme voisins.

$\Omega = \Omega_1 \times \cdots \times \Omega_n$ est l'ensemble des observations jointes des agents, avec Ω_i les observations possibles de l'agent i. En faisant l'hypothèse que l'observation d'un agent est indépendante des autres agents, la fonction d'observation peut se décomposer en un produit de probabilités d'observations locales :

$$O(s,a,s',(\omega_1,\ldots\omega_n)) = \prod_{i=1}^{n} O_i((s_i,s_u),a_i,(s'_i,s'_u),\omega_i) \tag{2.21}$$

où $\omega_i \in \Omega_i$ est l'observation de l'agent i et la fonction O_i donne la probabilité d'observer ω_i si l'agent arrive dans l'état local (s'_i,s'_u) après avoir effectué l'action locale a_i depuis l'état (s_i,s_u).

Un exemple illustratif des ND-POMDP, appliqué au problème de capteurs en réseau (figure 2.6), a été proposé dans [Nair et al., 2005]. Le but d'un réseau de capteurs est de détecter et de suivre des cibles à l'intérieur de l'environnement surveillé. Chaque capteur peut observer des zones dans plusieurs directions (gauche/droite). Cependant, ces capteurs ne sont pas parfaits et ont une probabilité de donner des résultats incorrects. Afin de limiter ces erreurs, un capteur ne sera récompensé pour la détection d'une cible que si un autre capteur la détecte de façon simultanée. Dans ce cas, chaque agent impliqué obtiendra une forte récompense. Dans le cas contraire, ils ne seront pas récompensés. Cependant, l'observation d'une zone par un capteur a

FIGURE 2.6 – Scénario de réseau de capteurs ; 4 capteurs (cercles) et 3 zones (Locx-x)

un coût non nul, quelque soit le résultat de l'observation. Il est néanmoins possible de désactiver des capteurs afin de minimiser les coûts.

Dans ces réseaux, les cibles sont indépendantes. Les mouvements d'une cible ne dépendent ni des autres cibles ni des capteurs. Ces déplacements sont incertains. Il n'est donc pas possible de connaître avec certitudes la prochaine position de la cible.

L'objectif consiste donc à coordonner l'utilisation des différents capteurs afin de minimiser le nombre de capteurs observant une zone tout en maximisant le nombre de capteurs qui observent les cibles.

Un tel problème correspond à un ND-POMDP, où chaque capteur correspond à un agent. L'état du système correspond à la position de chaque cible dans l'environnement. Comme la position des cibles est indépendante des agents, on a :

$$S_i = \emptyset, \forall i \in Ag \tag{2.22}$$

$$S_u = S_{cible_1} \times \cdots \times S_{cible_z} \tag{2.23}$$

où, S_{cible_i} représente les positions possibles de la cible i dans l'environnement. Dans le cas du réseau de la figure 2.6, les états possibles d'une cible sont $S_{cible_i} \subseteq$ {absent, Loc1-1, Loc2-1, Loc2-2}. Similairement, les actions d'un agent/capteur seront $A_i \subseteq$ {observer à gauche, observer à droite, désactivation}.

La fonction de récompense sera décomposée en deux parties : les récompenses individuelles qui représentent le coût d'utilisation de chaque capteur, et les récompenses de zone. Pour chaque zone, si au moins deux agents la surveillent et qu'une cible s'y trouve, ils obtiennent une récompense, sinon rien.

La fonction de transition d'un capteur sera définie en fonction des probabilités de faux positifs et de faux négatifs. La fonction de transition sera définie uniquement sur les distributions de probabilité de déplacement de chaque cible.

Reprenons le problème du « Coffee Robot » et considérons que deux robots doivent chercher du café pour leur humain respectif. L'évolution des robots dans l'environnement peut être considérée comme totalement indépendante. Cependant, si les robots ne livrent pas simultanément les cafés aux humains, l'humain le recevant en dernier ne sera pas aussi satisfait. La fonction de récompense est donc la combinaison d'une récompense locale à chaque robot et globale. La récompense locale est la récompense précédemment définie : le robot est sec et l'humain a du

café. La récompense globale contiendra une pénalité tant qu'un humain est satisfait et qu'il existe au moins un humain sans café.

Cette modélisation peut se rapprocher d'un problème d'optimisation de contraintes, dont les variables sont les politiques de chaque agent et dont les contraintes sont les fonctions de récompense. Pour cette raison, l'algorithme LID-JESP s'inspire d'une méthode distribuée de satisfaction de contraintes : le « Distributed Breakout Algorihtm » [Yokoo and Hirayama, 1996].

Distributed Breakout Algorithm

Le Distributed Breakout Algorithm (DBA) [Yokoo and Hirayama, 1996] est un algorithme distribué de satisfaction de contraintes (CSP). Il s'inspire du Breakout Algorithm [Morris, 1993] qui est défini pour des problèmes centralisés. Le DBA a pour but de maximiser le nombre de contraintes satisfaites par les agents.

Pour rappel, un CSP est composé de n variables $x_1, \ldots, x_n$ à domaines finis : $D_1, \ldots, D_n$, ainsi que d'un ensemble fini de contraintes sur les variables. Une contrainte est un prédicat $p_k(x_{k_1}, \ldots, x_{k_j})$ défini sur $D_{k_1} \times \cdots \times D_{k_j}$, avec $j \leq n$. Une contrainte n'est satisfaite que si l'assignation des variables $(x_{k_1}, \ldots, x_{k_j})$ vérifie le prédicat p_k.

La distribution d'un CSP consiste à répartir les variables entre les différents agents. Chaque agent devra donc assigner une valeur à chacune de ses variables de façon à satisfaire un nombre maximal de contraintes. Les contraintes pouvant s'appliquer sur plusieurs variables, celles-ci pourront être dépendantes de plusieurs agents. Deux agents seront considérés comme voisins, s'il existe une même contrainte affectant des variables contrôlées par ces deux agents.

L'algorithme consiste à partir d'une assignation initiale arbitraire des variables de chaque agent et à améliorer localement et itérativement la satisfaction du problème. Pour éviter les changements conflictuels, deux voisins ne pourront modifier leurs assignations en même temps. Il s'agit alors de coévolution. Chaque agent échangera, avec son voisinage, une évaluation pondérée des violations de contraintes dont il est responsable ainsi que l'amélioration qu'il peut effectuer grâce à une réassignation. Seuls les agents qui proposent la plus grande amélioration dans leur voisinage pourront mettre à jour leurs affectations. Dans le cas où un agent ne satisfait pas toutes ces contraintes et qu'aucun agent ne peut s'améliorer, alors les pondérations des contraintes non satisfaites seront modifiées, afin d'éviter les optimums locaux.

L'algorithme possède un mécanisme distribué de détection de solution et pourra donc s'arrêter automatiquement lorsqu'une solution est trouvée. Ce mécanisme est présenté dans la section suivante.

LID-JESP

LID-JESP [Nair et al., 2005] est un algorithme distribué fortement inspiré par le DBA [Yokoo and Hirayama, 1996], appliqué à la résolution des ND-POMDPs. Avec cet algorithme, chaque

agent va essayer d'améliorer sa politique en respectant celles reçues de son voisinage, c'est-à-dire les agents avec qui il partage des contraintes représentées par des récompenses.

Les agents commencent avec une politique quelconque, qu'ils échangeront avec leurs voisins. En utilisant les politiques reçues de son voisinage, chaque agent peut calculer l'amélioration possible de sa politique locale, en mesurant l'évaluation des états avec les actions jointes possibles, et l'échanger avec ces derniers. Si l'amélioration locale de l'agent est la plus grande du voisinage lors de l'itération, alors celui-ci mettra à jour sa politique, pour profiter de cette amélioration, puis la partagera avec ses voisins. Un principe similaire de coévolution des politiques est utilisé par [Chades et al., 2002] pour traiter des Dec-POMDPs. Les itérations continueront jusqu'à ce qu'une solution localement optimale soit atteinte, auquel cas l'algorithme s'arrêtera dès que les agents auront détecté la convergence.

La convergence globale est détectée localement grâce à l'utilisation d'un compteur par chaque agent. Celui-ci est initialisé à la distance maximale entre deux agents. Deux voisins sont à une distance de 1 et le voisin d'un voisin est à une distance maximale de 2. Ce compteur diminue de 1 lorsqu'il n'existe pas d'amélioration à la politique locale actuelle ou est réinitialisé dans les autres cas. Une fois le compteur mis à jour, il est échangé avec les voisins et reçoit la plus grande valeur du voisinage. Lorsque son compteur atteint 0, chaque agent termine l'algorithme. L'utilisation de ce type de compteur garantit l'arrêt de la résolution si et seulement si aucun agent ne peut améliorer sa solution locale [Nair et al., 2005].

2.5 Conclusion

La planification de systèmes stochastiques à l'aide de processus décisionnels markoviens a été présentée dans ce chapitre. Une des principales limitations de ce type de modélisation est l'incapacité à traiter des problèmes de tailles conséquentes. Cette incapacité est due à une croissance exponentielle de la représentation en fonction du nombre d'états et d'actions.

Des extensions multi-agents ou prenant en compte les capacités d'observation existent dans la littérature pour augmenter les capacités de modélisation. Cependant, ces modélisations plus larges ont de plus grandes difficultés à passer à l'échelle. De ce fait, différentes techniques de modélisation et de résolution permettant une réduction de la complexité ont été présentées.

Un premier ensemble de méthodologies pour augmenter la scalabilité des modélisations a été décrit. Les MDPs factorisés utilisent une représentation compacte des fonctions de transition et de récompense. Pour cela, ils exploitent la divisibilité des états en variables dont les évolutions sont indépendantes. Les approches décomposées divisent les problèmes en sous-problèmes de façon à ce que ces sous-problèmes soient faiblement connectés. Ces nouveaux problèmes sont ainsi plus simples à résoudre. Une solution globale pourra être obtenue à partir des solutions obtenues pour chaque sous-problème. Finalement, les méthodes basées Monte-Carlo ne nécessitent pas de représenter explicitement le modèle, mais requièrent un simulateur. Grâce à une suite de

simulations semi-aléatoires, une fonction de valeur approximée pourra être utilisée pour obtenir une politique.

Un second type de méthodologies répartit la charge de calcul sur plusieurs agents. La modélisation par ND-POMDPs bénéficie de l'indépendance des transitions de chaque agent pour séparer le modèle. L'algorithme LID-JESP permet d'obtenir une solution pour ce type de modèle grâce à la coévolution des politiques. L'exploitation d'un faible nombre d'interactions entre les agents est une solution utilisée pour séparer le problème en zones avec et sans interaction. Cette séparation permet ainsi de réduire la complexité du problème à résoudre par chaque agent.

Dans le chapitre suivant, les limites de ces approches par rapport à la problématique traitée par cette thèse seront discutées.

Chapitre 3

Modélisation de la gestion d'une ressource partagée dans un réseau

Ce chapitre introduit une classe de problèmes d'optimisation de la gestion d'une ressource partagée dans des réseaux complexes sur un horizon de prédiction en tenant compte des incertitudes. Cette optimisation doit prendre en compte l'ensemble des incertitudes affectant les réseaux considérés afin de rendre résiliente leur gestion.

Dans une majorité d'applications réelles, les états d'un système et les actions l'affectant ne sont pas des ensembles finis ; le volume d'une rivière et la vitesse d'un robot sont des variables continues. De ce fait, les MDPs et leurs extensions ne permettent pas de modéliser directement ces applications. Des variantes permettant de modéliser des ensembles continus d'états [Marecki et al., 2006] ou d'actions [Mansley et al., 2011] ont été définies. Néanmoins, plutôt que d'utiliser des MDPs continus, la solution consiste souvent à discrétiser les espaces d'états et d'actions.

Une nouvelle approche permettant la modélisation de la gestion prédictive sous incertitudes d'une ressource continue et partagée sur un réseau est ainsi présentée dans ce chapitre. Cette approche est illustrée sur la problématique de gestion des voies navigables. Dans un second temps, l'applicabilité des méthodes existantes pour la problématique considérée est discutée. Cette modélisation reprend des travaux présentés dans [Desquesnes et al., 2016c] et [Desquesnes et al., 2017a].

3.1 Classe de problèmes

Nous nous plaçons dans le cadre du problème de gestion prédictive sous incertitudes d'une ressource partagée sur un réseau de grande taille. Les systèmes considérés sont constitués de composants interconnectés en réseau. Ces composants disposent d'une ressource commune qui doit être gérée de façon efficiente au cours du temps afin de répondre à des objectifs de gestion. La ressource doit être partagée entre chaque composant du réseau. Il s'agit donc de répondre aux

objectifs de chaque composant individuellement, tout en garantissant le respect des objectifs de gestion du réseau complet. Des effecteurs (ou actionneurs) permettent le partage/déplacement de cette ressource entre les composants. Ce partage est cependant contraint de par les quantités de ressource transférables et par les connexions entre les composants. La classe de problèmes est définie pour la gestion d'une ressource continue, mais est applicable à des problèmes de gestion discrets.

3.1.1 Définition du problème

Les effecteurs relient les composants entre eux et permettent ainsi le déplacement de la ressource. Chaque effecteur peut prendre de la ressource dans un ou plusieurs composants sources afin de la déplacer dans un ou plusieurs composants cibles. Le déplacement de ressource par un effecteur est limité par sa capacité de transfert. De plus, les transferts peuvent être affectés par des variations ou des perturbations stochastiques, connues ou non, affectant les effecteurs à un moment donné. Sauf indications contraires, les déplacements de ressource sont considérés simultanés et instantanés, en considérant des pas de temps relativement larges du point du vue des problèmes traités.

Un composant est une partie du réseau stockant une quantité de ressource. Chaque composant doit maintenir la ressource qu'il stocke dans une plage de fonctionnement. Le maintien de cette plage de fonctionnement est l'objectif principal de chaque composant. Lorsqu'un composant ne respecte plus sa plage de fonctionnement, celui-ci ne pourra plus réaliser sa tâche dans le système ce qui induira des coûts supplémentaires. Néanmoins, le composant pourra toujours participer aux partages de ressource avec les autres composants.

3.1.2 Représentation formelle

Ce problème peut ainsi être représenté par un graphe orienté fini $G = (E,L)$. L'ensemble $E = \{e_0,e_1,\ldots,e_n\}$ des nœuds du graphe représente les n composants du réseau modélisé. Un nœud supplémentaire e_0 est ajouté pour représenter les échanges possibles de ressource avec l'extérieur du modèle. Il permet ainsi une entrée et une sortie contrôlée de ressource dans le réseau modélisé. $L = \{l_1,\ldots,l_m\}$ est l'ensemble des m arêtes du graphe qui correspondent aux différentes connexions disponibles entre les composants et avec le modèle extérieur grâce aux effecteurs.

Des plages de fonctionnement sont définies pour les nœuds et représentent les quantités de ressource permettant à chaque composant de respecter ses objectifs. Pour un composant e_i, avec $i \in \{1,\ldots,n\}$, sa plage de fonctionnement est définie par l'intervalle $[I^-(e_i),I^+(e_i)]$, où la borne $I^+(e_i)$ (resp. $I^-(e_i)$) est la quantité maximale (resp. minimale) de ressource pouvant être stockée dans le nœud e_i au temps t. e_0 correspondant à l'extérieur du modèle, aucune plage de fonctionnement ne lui est associée. Similairement, à chaque arête l_i est associée une plage de quantité de ressource pouvant être transférée par cette arête : $[I^-(l_i),I^+(l_i)]$. Les capacités

de transfert et de stockage peuvent évoluer au cours du temps. Cependant, par simplicité de notation elles seront considérées comme constantes. La figure 3.1 donne un exemple de graphe correspondant à la classe de problèmes ici définie, où quatre composants peuvent faire circuler une ressource entre eux, mais aussi l'échanger avec un réseau extérieur. Si le graphe non orienté privé de e_0 n'est pas connexe, c'est-à-dire s'il existe deux nœuds e_i et e_j qui ne peuvent être reliés sans passer par le nœud e_0, alors le système pourra être divisé en plusieurs réseaux indépendants.

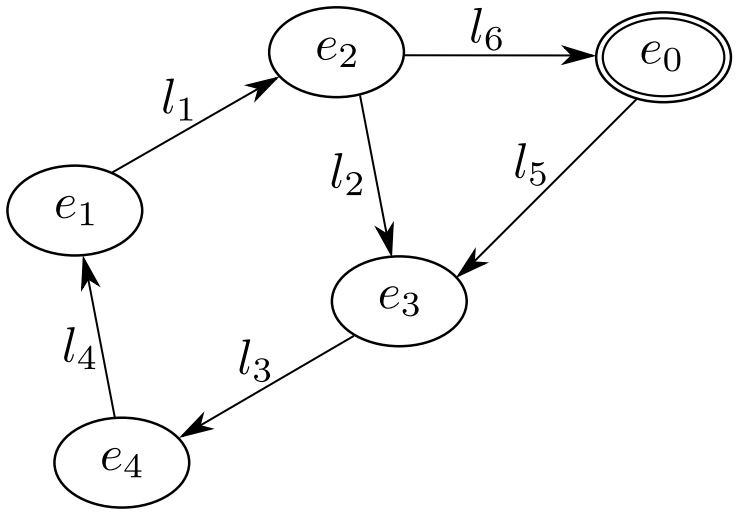

FIGURE 3.1 – Exemple de modèle correspondant à la classe de problèmes : 4 composants (e_1, e_2, e_3 et e_4) ayant des échanges entre eux via des effecteurs (l_1, l_2, l_3 et l_4). Des effecteurs (l_5 et l_6) relient le système modélisé à un système non modélisé (e_0)

3.1.3 Objectifs

Le respect de la plage de fonctionnement de chaque composant est l'objectif principal du problème de gestion. C'est-à-dire, pour tout temps t et pour tout composant e_i, avec $i \in \{1, \ldots, n\}$, il faut que :

$$I^-(e_i) \leq Q_t(e_i) \leq I^+(e_i) \tag{3.1}$$

où $Q_t(e_i)$ est la quantité de ressource stockée au temps t dans le composant e_i.

Néanmoins, le modèle permet en toute circonstance le non-respect de cette plage. Par exemple, bien qu'un composant n'ait plus de ressource à un temps donné, il est toujours possible qu'il fournisse des ressources, même sans réalité physique. Ces violations de la plage de fonctionnement seront limitées par un choix judicieux des fonctions de coût associées à chaque composant. Un coût infini lors de l'atteinte d'un tel état aura un effet équivalent à son interdiction. Le choix de permettre des violations de la plage de fonctionnement est fait dans le but de simplifier la définition de la fonction de transition, en autorisant toutes combinaisons de ressource déplacées depuis chaque état du système.

Chaque composant possède également une fonction de coût f_{e_i} qui pénalise l'écart de la quantité de ressource qu'il possède à un instant t : $Q_t(e_i)$, par rapport à une quantité optimale : $Q^*(e_i)$. D'autres coûts peuvent être présents sur le réseau selon les problèmes traités, tel qu'un coût de déplacement de ressource défini pour chaque effecteur par :

$$g_{l_i} : [I^-(l_i),I^+(l_i)] \rightarrow \mathbb{R}^+ \tag{3.2}$$

3.1.4 Exemples d'applications

Cette classe de problèmes regroupe différentes applications, telles que la gestion de flottes de véhicules de location ou de répartition d'énergie entre différents points de stockage. Dans le cas du problème de gestion de flottes de véhicules, le but sera de répartir les véhicules (la ressource) entre différents gares (les composants) de façon à pouvoir anticiper aux mieux les besoins des clients avec un moindre coût. Chaque gare aurait donc une plage de fonctionnement définie par le nombre maximum de véhicules pouvant être stockés et un seuil minimum de sécurité. Le déplacement de véhicules (les effecteurs) à l'intérieur du réseau sera utilisé par les gestionnaires pour réguler la ressource (nombre de véhicules) de chaque entrepôt. La ressource peut entrer ou sortir du réseau modélisé lors des locations et des retours des véhicules. Dans cet exemple, déplacer la ressource aura un coût non négligeable.

Similairement pour un problème de gestion d'énergie, le but serait de répartir l'énergie (la ressource) entre différentes batteries (les composants) par des lignes électriques (les effecteurs) tout en prenant en compte une estimation de la consommation des utilisateurs (le réseau extérieur). Dans ces deux exemples, il est important de préciser qu'il s'agit de définir une stratégie de répartition de la ressource en prenant en compte une demande estimée des utilisateurs.

Afin de pouvoir être modélisé par un processus décisionnel markovien, il est nécessaire de définir un ensemble fini d'états et d'actions. Ces états et actions permettront de représenter le modèle ainsi que ses capacités d'évolutions.

3.2 Définition des états et des actions

Dans cette classe de problèmes, un état du système est défini par la quantité de ressource présente dans chaque composant considéré à un instant donné, $Q_t(e_i)$. De ce fait, l'état s du système à un temps t peut s'écrire sous la forme suivante :

$$s = \{Q_t(e_1), \ldots, Q_t(e_n)\} \tag{3.3}$$

Les actions représentent les combinaisons possibles de déplacement de ressource entre les composants grâce aux effecteurs du réseau. Elles sont considérées comme indépendantes du temps. Une action a définit la quantité de ressource qui sera déplacée par chaque effecteur lorsqu'elle sera appliquée. Elle peut donc s'écrire sous la forme :

$$a = \{Q(l_1), \ldots, Q(l_m)\} \tag{3.4}$$

où $Q(l_i) \in [I^-(l_i),I^+(l_i)]$ est la quantité de ressource qui sera déplacée par le i$^{\text{ème}}$ effecteur.

Avec ses définitions sur des domaines continus, il existe une infinité d'états et d'actions pouvant être définis. Or, les formalismes basés sur les MDPs s'accommodent d'avantage d'un ensemble fini d'états et d'actions. Pour cela, une discrétisation des états et des actions est proposée.

3.2.1 Discrétisation des actions

L'ensemble des quantités possibles transférées par un effecteur l_i sera donc discrétisé en un ensemble fini d'intervalles A_{l_i} :

$$A_{l_i} = \{a^1_{l_i}, \ldots, a^k_{l_i}\} \tag{3.5}$$

où k est le nombre de subdivisions de cet effecteur. $a^j_{l_i}$ est un intervalle de quantité transférable par l'effecteur l_i. L'union des intervalles est égale à la capacité de transfert possible par l'effecteur :

$$\bigcup_{j=1}^{k} a^j_{l_i} = [I^-(l_i),I^+(l_i)] \tag{3.6}$$

De plus, les intervalles ainsi produits ne se chevauchent pas et sont nommés par ordre croissant. Pour chaque effecteur l_i et pour toute paire d'intervalles $a^x_{l_i}$ et $a^y_{l_i}$, tel que $x < y$:

$$\forall q_x \in a^x_{l_i}, \forall q_y \in a^y_{l_i} \qquad q_x < q_y \tag{3.7}$$

où q_x (resp. q_y) est une quantité possible de ressource transférée par l'effecteur l_i lorsque l'intervalle $a^x_{l_i}$ (resp. $a^y_{l_i}$) est utilisé.

Ainsi l'ensemble fini des actions est défini par la combinaison des intervalles correspondant à chaque effecteur :

$$A = \prod_{i=1}^{m} A_{l_i} \tag{3.8}$$

où m est le nombre d'effecteurs.

Par exemple sur un réseau de batteries, un câble l_i permet de déplacer l'énergie stockée d'une batterie à une autre. Sur un pas de temps donné, il peut déplacer entre 0 et 30 Coulomb. Ainsi $I^-(l_i) = 0$ et $I^+(l_i) = 30$. Si cette plage est divisée en trois intervalles régulier alors :

$$a^1_{l_i} = [0,\ 10[\qquad a^2_{l_i} = [10,\ 20[\qquad a^3_{l_i} = [20,\ 30] \tag{3.9}$$

$$A_{l_i} = \{a^1_{l_i},a^2_{l_i},a^3_{l_i}\} \tag{3.10}$$

3.2.2 Discrétisation des états

Similairement, l'ensemble des quantités de ressources pouvant être stockées dans un composant doit être discrétisé. Cependant, contrairement aux effecteurs, cet ensemble n'est pas limité

à la plage de fonctionnement du composant. De ce fait, en plus d'une division de la plage de fonctionnement, deux autres intervalles sont considérés pour englober les valeurs hors de la plage de fonctionnement normale. Ainsi, l'état interne du composant e_i est défini par l'ensemble fini d'intervalles S_{e_i} tel que :

$$S_{e_i} = \{s^0_{e_i}, s^1_{e_i}, \ldots, s^k_{e_i}, s^{k+1}_{e_i}\} \tag{3.11}$$

où $s^j_{e_i}$ est un intervalle de quantité stockée dans le composant e_i. Les intervalles de $s^1_{e_i}$ à $s^k_{e_i}$ contiennent l'intégralité des valeurs correspondant au fonctionnement du composant, c'est-à-dire :

$$[I^-(e_i), I^+(e_i)] \subseteq \bigcup_{j=1}^{k} s^j_{e_i} \tag{3.12}$$

Les deux autres intervalles, $s^0_{e_i}$ et $s^{k+1}_{e_i}$, sont deux intervalles de taille infinie. Ils permettent de représenter le non-respect de la plage de fonctionnement du composant. Selon les problèmes traités, il peut être intéressant de considérer que les k intervalles $\{1, \ldots, k\}$ dépassent légérement de la plage de fonctionnement. Ceci rend possible une modélisation plus fine des violations des contraintes de fonctionnement, puisqu'il existe des intervalles de tailles finies représentant leurs non respect. Comme pour les actions, ces intervalles sont ordonnés de façon croissante et ne se chevauchent pas.

Ainsi, l'ensemble fini des états après la discrétisation est défini par la combinaison des intervalles de chaque composant et du temps :

$$S = [0, \ldots, \tau] \times \prod_{i=1}^{n} S_{e_i} \tag{3.13}$$

où τ est le nombre de pas de temps pris en compte par la modélisation. La prise en compte du temps dans l'état est nécessaire, car les perturbations affectant le système peuvent dépendre du temps.

Reprenons l'exemple du réseau de batteries. Une batterie e_i doit stocker entre 10 et 70 Coulomb pour garantir le bon fonctionnement du réseau. Ainsi les bornes de sa plage de fonctionnement sont $I^-(e_i) = 10$ et $I^+(e_i) = 70$. Si la décomposition est réalisée en intervalles de taille 20, alors une décomposition possible est :

$$S_{e_i} = \{\ ]-\infty, 10[,\ [10, 30[,\ [30, 50[,\ [50, 70],\ ]70, +\infty[\} \tag{3.14}$$

où le premier et le dernier intervalles correspondent à l'ensemble des valeurs hors de la plage de fonctionnement.

Un second type de discrétisation est :

$$S_{e_i} = \{\ ]-\infty, 0[,\ [0, 20[,\ [20, 40[,\ [40, 60[,\ [60, 80],\ ]80, +\infty[\} \tag{3.15}$$

Ce type de discrétisation permet de mieux représenter des écarts faibles à la plage de fonctionnement, grâce aux intervalles chevauchant ses bornes. Contrairement à la première discrétisation qui ne fait pas de différence entre un écart mineur et majeur à la plage de fonctionnement. Le type de discrétisation à utiliser est lié au problème traité.

3.2.3 Définition de la fonction de coût

L'objectif de ce type de problème est de maintenir la quantité de ressource de chaque composant dans sa plage de fonctionnement tout en minimisant l'écart de cette ressource à une valeur optimale. Une pondération peut être présente sur les effecteurs afin de favoriser l'utilisation de certains d'entre eux. La définition des états et des actions permet d'introduire la fonction de coût du problème $C(s,a,s')$, avec $s,s' \in S$ et $a \in A$, correspondant à ces objectifs :

$$C(s,a,s') = \sum_{e \in E} f_e(s'_e) + \sum_{l \in L} g_l(a_l) = -R(s,a,s') \tag{3.16}$$

où f_e et g_l sont les fonctions qui évaluent respectivement la quantité de ressource stockée dans le composant e et transférée par l'effecteur l. La fonction f_e doit pénaliser fortement le non-respect de la plage de fonctionnement et tout particulièrement le fait d'atteindre un des intervalles de taille infinie. $a_l \in A_l$ est l'intervalle de ressource transférée par l'effecteur l lorsque l'action a est appliquée. Similairement $s'_e \in S_e$ est l'intervalle de ressource stockée par le composant e lorsque l'état s' est atteint. Ces fonctions seront à définir par les opérateurs en fonction des problèmes traités. La fonction f_e a pour but de garantir le bon fonctionnement du système tandis que g_l visera à optimiser sa gestion.

3.3 Évolution du système sous incertitudes

L'évolution de l'état de chaque composant dépend principalement de son état actuel et des quantités de ressource qui sont transférées par les effecteurs auxquels il est connecté. Cependant, cette évolution dépend aussi des différentes incertitudes qui peuvent affecter le système modélisé. Deux types d'incertitudes peuvent être définis : les incertitudes du modèle qui découlent de la connaissance du système à contrôler et les incertitudes de la discrétisation des états et des actions en intervalles.

3.3.1 Incertitudes liées au modèle

Le transfert de ressource par un effecteur n'est pas forcément déterministe. Plusieurs perturbations peuvent affecter la quantité de ressource échangée, telles qu'une perte de ressource lors du transfert due par exemple à une fuite, ou encore à une imprécision dans la mesure de la ressource déplacée résultant en un transfert plus ou moins important, que prévu. De même, des perturbations peuvent affecter distinctement un ou plusieurs composants du réseau. Par exemple, des véhicules peuvent tomber en panne dans un centre de location et ainsi devenir indisponibles. Un autre exemple serait un même événement pluvieux affectant le niveau de plusieurs biefs.

Ces perturbations sont temporalisées, c'est-à-dire qu'une perturbation est définie pour un instant donné. Dans le modèle proposé, les perturbations dont l'occurrence est conditionnée par l'occurrence préalable d'une autre perturbation ne sont pas considérées. Par exemple, il ne serait

pas possible de considérer un événement de pluie affectant le réseau modélisé sur 2 pas de temps. Ces perturbations devraient être représentées de façon indépendante ou via une modification de la définition des états. Ceci est nécessaire afin que l'évolution d'un état ne dépende pas du passé.

Par simplicité et sans perte d'information, il est possible de définir une perturbation comme affectant uniquement les effecteurs. Celles touchant aux composants peuvent être vues comme des perturbations sur des effecteurs reliant ces composants au réseau non modélisé (e_0).

Ainsi chaque effecteur l possède un ensemble fini de variations $\mathcal{V}_t(l)$, potentiellement discrétisées, pouvant survenir au pas de temps t. Il est possible que les variations de plusieurs effecteurs soient interdépendantes. L'ensemble des variations pouvant affecter les différents effecteurs est noté $\mathcal{V}$ et est défini par :

$$\mathcal{V} = \prod_{t=0}^{\tau} \prod_{i=1}^{m} \mathcal{V}_t(l_i) \tag{3.17}$$

Soit $v \in \mathcal{V}$ une variation qui affecte le réseau à un instant donné. La notation $P(v|s,a)$ représente la probabilité que la variation v affecte le réseau lorsque celui-ci se trouve dans un état s et effectue une action a.

Ces incertitudes ne sont cependant pas les seules qui affectent le problème. La discrétisation des états et actions en intervalles est aussi source d'incertitudes.

3.3.2 Incertitudes liées à la discrétisation

La discrétisation des états et des actions en intervalles de quantités induit une double source d'incertitudes. L'état d'un composant est représenté par un intervalle de ressource. Il est donc impossible de déterminer quelle est la quantité réelle de ressource dans cet intervalle. De même, l'action d'un effecteur correspond à un intervalle de ressource déplacé dans ce réseau, ce qui ne permet pas de connaître précisément la quantité de ressource réellement déplacée.

La discrétisation des états et des actions a donc un impact sur la modélisation de l'évolution du système par la fonction de transition. Son impact peut cependant être atténué en faisant des hypothèses sur la répartition des valeurs dans les intervalles. Par simplicité, il sera considéré que les quantités représentées par un intervalle d'un composant sont uniformément réparties. La même hypothèse est faite lors du bilan du déplacement de ressource de ce composant. Ce raisonnement peut néanmoins être appliqué en utilisant d'autres types de répartition.

Ainsi, soit $e \in E$ un composant du réseau et $s \in S$ l'état du système à un temps t. L'intervalle de ressource du composant e de cet état s est défini par $s_e = [s_e^-, s_e^+]$, où s_e^- et s_e^+ sont respectivement les bornes minimales et maximales de l'intervalle s_e. Soit $a \in A$ une action et $v \in \mathcal{V}$ une variation, alors $av_e = [av_e^-, av_e^+]$ est défini comme l'intervalle des changements possibles de ressource locale du composant e après l'application de l'action a lorsqu'une perturbation impliquant une variation v se produit. La valeur attendue après avoir effectué une action est définie par la somme de la valeur moyenne de l'intervalle d'état local s_e et la moyenne des échanges liés

aux actions locales et aux variations locales av_e.

$$\overline{s_e} + \overline{av_e} \tag{3.18}$$

où l'opérateur $\overline{\ }$ retourne la valeur moyenne de l'intervalle. L'intervalle $s_e^* = [s_e^{*-}, s_e^{*+}]$ est défini comme l'intervalle contenant la valeur attendue :

$$\overline{s_e} + \overline{av_e} \in s_e^* \tag{3.19}$$

En utilisant ces variables, il est possible de représenter graphiquement l'ensemble des quantités de ressource qui peuvent être atteintes, au pas de temps suivant, pour chaque valeur de l'intervalle de départ (figure 3.2). Ces quantités possibles sont représentées par la zone entre les droites :

$$min(x) = x + av_e^- \tag{3.20}$$
$$max(x) = x + av_e^+ \tag{3.21}$$

Ces droites représentent respectivement la quantité minimale et maximale de ressource pouvant être atteinte pour chaque état réel possible du composant au pas de temps précédent.

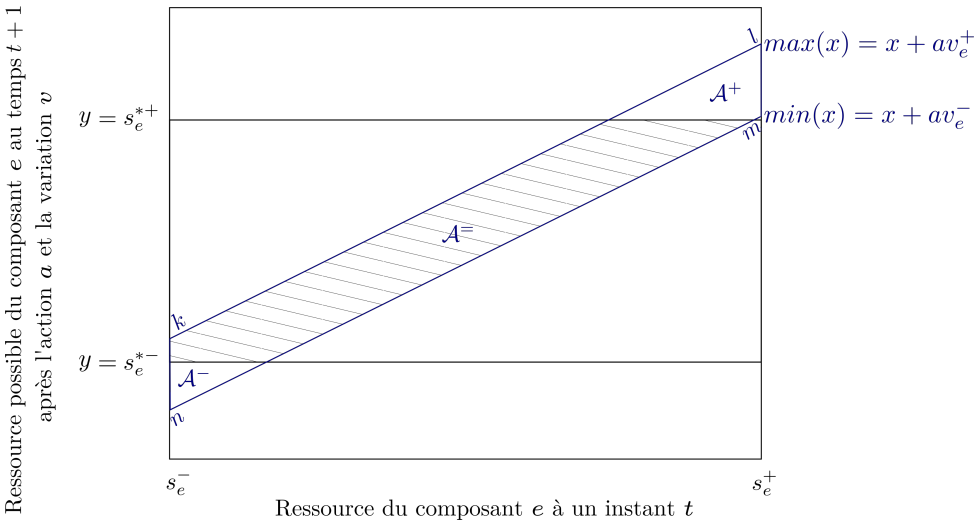

FIGURE 3.2 – Effet de la discrétisation en intervalle sur la transition. L'intervalle de départ est en abscisse et les volumes résultants possibles en ordonnée

Ainsi, la probabilité d'atteindre l'intervalle attendu s_e^* au temps $t+1$ après avoir effectué l'action a sous la variation v depuis l'intervalle s_e au temps t correspond au pourcentage de l'aire entre les droites $y = min(x)$ et $y = max(x)$ bornée dans l'intervalle s_e^*. Ceci correspond à la zone hachurée de la figure 3.2.

Ce raisonnement peut être appliqué pour déterminer la probabilité d'atteindre chaque intervalle du composant. Le nombre d'intervalles atteignables dépend des discrétisations des états et des actions. Les discrétisations ont été choisies de façon à ce que seuls trois intervalles puissent être atteints pour chaque couple d'action et de variation, ce qui simplifie la rédaction et les calculs. Ces intervalles sont définis à partir des bornes de l'intervalle attendu :

— l'intervalle attendu : $[s_e^{*-}, s_e^{*+}]$;

— l'intervalle supérieur : $[s_e^{*+}, s_e^{*++}]$;

— l'intervalle inférieur : $[s_e^{*--}, s_e^{*-}]$.

La relation entre ces intervalles est représentée sur la figure 3.3

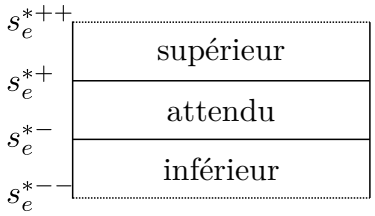

FIGURE 3.3 – Relation entre les états attendu, supérieur et inférieur

Pour déterminer la probabilité d'atteindre chaque intervalle, quatre points k, l, m et n sont définis. Ils correspondent aux valeurs minimales et maximales pouvant être atteintes via les bornes de l'intervalle de départ.

$$\begin{aligned} k &= (s_e^-, s_e^- + av_e^+) & l &= (s_e^+, s_e^+ + av_e^+) \\ n &= (s_e^-, s_e^- + av_e^-) & m &= (s_e^+, s_e^+ + av_e^-) \end{aligned} \tag{3.22}$$

Le parallélogramme $klmn$ représente l'ensemble des valeurs possibles du composant e au temps $t+1$ après avoir effectué l'action a avec la variation v depuis l'état s au temps t. De ce fait, la probabilité que l'intervalle suivant soit l'intervalle attendu correspondant au pourcentage que représente l'aire de $klmn$ par rapport au rectangle délimité par $y = s_e^{*-}$ et $y = s_e^{*+}$.

L'aire du parallélogramme $\mathcal{A}$ peut ainsi être divisée en trois parties $\mathcal{A}^-$, $\mathcal{A}^=$ et $\mathcal{A}^+$ (visible sur la figure 3.2), tel que $\mathcal{A} = \mathcal{A}^- + \mathcal{A}^= + \mathcal{A}^+$ où :

— A^- est l'aire de $klmn$ sous $y = s_e^{*-}$, c-à-d l'intervalle inférieur ;

— $A^=$ est l'aire de $klmn$ entre $y = s_e^{*-}$ et $y = s_e^{*+}$, c-à-d l'intervalle attendu ;

— A^+ est l'aire de $klmn$ au dessus de $y = s_e^{*+}$, c-à-d l'intervalle supérieur.

Les probabilités d'atteindre chaque intervalle sont définies comme des ratios d'aire :

$$p^- = \frac{\mathcal{A}^-}{\mathcal{A}} \qquad p^= = \frac{\mathcal{A}^=}{\mathcal{A}} \qquad p^+ = \frac{\mathcal{A}^+}{\mathcal{A}} \tag{3.23}$$

L'aire du parallélogramme $klmn$ s'écrit :

$$\mathcal{A} = (s_e^+ - s_e^-) \times (av_e^+ - av_e^-) \tag{3.24}$$

Comme partiellement visible sur la figure 3.2, les parties de $klmn$ hors de s_e^* sont de forme triangulaire ou trapézoïdale selon les intersections des droites. Il est ainsi possible d'écrire :

$$\mathcal{A}^+ = \begin{cases} \dfrac{((av_e^+ - s_e^{*+} + s_e^+)^2 - (av_e^- - s_e^{*-} + s_e^-)^2)}{2} & \text{si} \quad min(s_e^+) > s_e^{*+} \\ \dfrac{(av_e^+ - s_e^{*+} + s_e^+)^2}{2} & \text{si} \quad max(s_e^+) > s_e^{*+} \wedge min(s_e^+) \leq s_e^{*+} \\ 0 & \text{sinon} \end{cases} \tag{3.25}$$

$$\mathcal{A}^- = \begin{cases} \dfrac{((av_e^- - s_e^{*-} + s_e^-)^2 - (av_e^+ - s_e^{*+} + s_e^+)^2)}{2} & \text{si} \quad max(s_e^-) < s_e^{*-} \\ \dfrac{(av_e^- - s_e^{*-} + s_e^-)^2}{2} & \text{si} \quad min(s_e^-) < s_e^{*-} \wedge max(s_e^-) \geq s_e^{*-} \\ 0 & \text{sinon} \end{cases} \tag{3.26}$$

avec les fonctions $min(x)$ et $max(x)$ définies par la relation 3.21.

La probabilité de changement d'intervalle est donc formulée par :

$$P(s'_e|s_e,av_e) = \begin{cases} p^=(s_e,av_e,s'_e) & \text{si} \quad s_e^* \in [s_e'^-,s_e'^+] \\ p^+(s_e,av_e,s'_e) & \text{si} \quad s_e^* \in [s_e'^{--},s_e'^-] \\ p^-(s_e,av_e,s'_e) & \text{si} \quad s_e^* \in [s_e'^+,s_e'^{++}] \\ 0 & \text{sinon} \end{cases} \tag{3.27}$$

3.3.3 Définition de la fonction de transition

L'évolution d'un composant dans le temps ne dépend que des effecteurs l'affectant et de son état actuel. Cependant, les variations présentes sur le réseau peuvent avoir un impact sur plusieurs composants en même temps ce qui implique nécessairement de calculer séparément la probabilité d'occurrence des variations et leurs impacts sur l'évolution des composants. La fonction de transition est donc définie comme la combinaison de la prise en compte des deux types d'incertitudes du modèle $P(v|s,a)$ et de la discrétisation $P(s'_e|s_e,av_e)$ par :

$$T(s,a,s') = \sum_{v \in \mathcal{V}} \left(P(v|s,a) \times \sum_{e \in E} P(s'_e|s_e,av_e) \right) \tag{3.28}$$

3.4 Modélisation d'un réseau de voies navigables

Ici, l'application ciblée de cette classe de problèmes est la gestion de la ressource en eau des voies navigables. L'objectif est de maintenir le niveau de chaque bief dans son rectangle de navigation tout en restant le plus proche d'un niveau idéal : le NNL. Ce niveau est principalement

perturbé par l'ouverture des écluses qui est requise pour le trafic fluvial. Des points de transferts tels que des écluses, des vannes ou encore des pompes, sont répartis entre les biefs pour permettre de réguler les niveaux d'eau. Finalement, ce réseau est connecté avec un ensemble de systèmes non modélisés tels que des rivières naturelles ou encore la mer.

En utilisant le modèle précédemment défini, il est possible de considérer les biefs comme des composants du système, dont le rectangle de navigation correspond à sa plage de fonctionnement, et les points de transfert comme des effecteurs. La ressource à optimiser pour chaque bief est donc l'eau. Bien que les points de transfert contrôlent un débit d'eau, par facilité de modélisation et de calculs, la quantité d'eau déplacée par les effecteurs et stockée dans les composants sera représentée de façon volumique. Cette modélisation a été introduite dans [Desquesnes et al., 2016c].

3.4.1 Définition des états et des actions

Un état du système de voies navigables doit décrire l'état de chaque bief et permettre de définir l'évolution future du réseau. Ce réseau est affecté par un grand nombre d'incertitudes liées au temps tels que l'influence de la météo, le trafic fluvial ou encore des rejets sauvages. Comme introduit précédemment, un état est défini par la combinaison d'un intervalle de volume décrivant chaque bief et par le pas de temps courant. Le choix a été fait de ne pas représenter l'état local d'un bief par un niveau d'eau, mais par le volume correspondant, dans le but de simplifier la modélisation des déplacements d'eau qui sont eux volumiques.

Les intervalles de volumes d'un bief sont définis autour de son rectangle de navigation, voir la figure 3.4. Pour un bief e, k_e intervalles sont utilisés pour découper le rectangle de navigation. Les intervalles 1 et k_e sont partiellement en dehors du rectangle de navigation. C'est-à-dire qu'ils contiennent respectivement des valeurs inférieures au LNL et supérieures au HNL. Inversement, les intervalles 0 et $k_e + 1$ sont entièrement en dehors du rectangle de navigation et sont donc de tailles infinies, afin de pouvoir représenter n'importe quel état possible du bief.

Ainsi l'ensemble des états du problème est défini par :

$$S = \{0, \dots, \tau\} \times \prod_{e \in E} \{0, \dots, k_e + 1\} \tag{3.29}$$

L'ensemble des actions correspond simplement à la discrétisation des volumes déplaçables par chaque point de transfert :

$$A = \prod_{l \in L} A_l \tag{3.30}$$

où l est un point de transfert orienté reliant deux biefs entre eux ou un bief à une partie du réseau non modélisée telle qu'une rivière naturelle ou la mer.

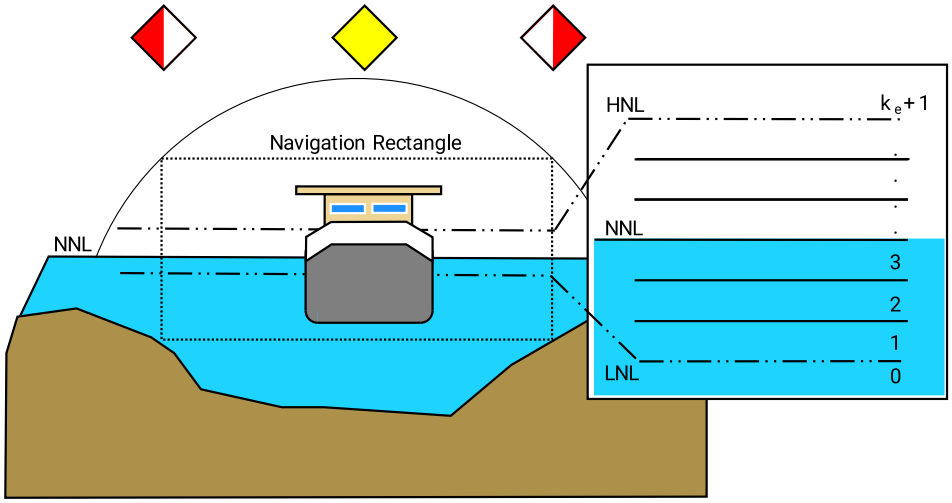

FIGURE 3.4 – Discrétisation des volumes d'un bief e en intervalles

3.4.2 Discrétisation des états et des actions

Une discrétisation possible des états et des actions en intervalles de volume peut se faire de façon à limiter le nombre d'états atteignables en réalisant une action et en fonction des variations incertaines (fuites, météo, ...). La combinaison d'une action et d'une variation peut à la fois faire baisser et monter le niveau d'un bief. De ce fait, les intervalles sont construits de telle sorte qu'une action a et une variation v ne peuvent conduire qu'à 3 intervalles possibles. Ceci implique que la discrétisation respecte pour chaque bief e et pour chaque couple d'action et de variation av :

$$\begin{aligned} s_e^+ + av_e^+ &\leq s_e^{*++} \\ s_e^- + av_e^- &\geq s_e^{*--} \end{aligned} \tag{3.31}$$

où $s_e^* = [s_e^{*-}, s_e^{*+}]$ est l'intervalle attendu, défini précédemment par $\overline{s_e} + \overline{av_e} \in s_e^*$. Les valeurs s_e^{*++} et s_e^{*--} correspondent respectivement aux bornes des intervalles supérieur et inférieur.

Ce type de discrétisation permet de limiter l'amplitude des actions par rapport à la définition des intervalles d'états en bornant les valeurs accessibles. D'autres discrétisations peuvent être proposées, permettant d'atteindre un plus grand nombre d'états après l'application de chaque couple d'action et de variation. Cependant, elles n'ont pas été définies puisqu'elles augmentent les incertitudes sur l'état réel, la vrai quantité de ressource stockée, atteint.

3.4.3 Fonction de coût

Le but de la gestion de l'eau dans un réseau de voies navigables est de maintenir le niveau de chaque bief dans son rectangle de navigation tout en étant le plus proche du niveau idéal. En pratique, le gestionnaire maintient le niveau de chaque bief très proche du NNL peu importe le

coût d'utilisation des points de transfert. La fonction de coût, qui est donc l'opposée de la fonction de récompense ($C(s,a,s') = -R(s,a,s')$) peut être définie en coopération avec des experts pour garantir l'optimalité des conditions de navigation sur l'ensemble du réseau de voies navigables :

$$C(s,a,s') = \sum_{e \in E} \frac{(\mathrm{NNL}_e - \overline{s'_e})^2}{(\mathrm{NNL}_e - w_e)^2} + \sum_{l \in L} g_l(a_l) \tag{3.32}$$

où NNL_e est le volume optimal correspondant au niveau normal de navigation du bief e. Lorsque s'_e, l'intervalle de volume du bief e dans l'état s', est hors du rectangle de navigation alors la valeur moyenne $\overline{s'_e}$ est remplacée par une grande valeur w_e. Dans le cas où s'_e est un intervalle partiellement hors du rectangle de navigation ou simplement lorsqu'il s'agit des intervalles bornant le rectangle de navigation (1 et k_e), alors une valeur $\frac{w_e}{2}$ est utilisée. Ces valeurs ont pour but de fortement pénaliser l'interruption de la navigation due au non-respect des conditions de navigation et de pénaliser des niveaux proches des bornes minimale et maximale définies par le rectangle de navigation.

g_l est une fonction à coût faible pour l'utilisation de chaque effecteur. Ces coûts sont définis en fonction des caractéristiques du réseau et des différents points de transferts. Cette fonction a pour but de privilégier les effecteurs les moins coûteux tels que les vannes qui permettent l'écoulement d'eau, plutôt que ceux potentiellement à fort coût tels que les pompes. g_l est négligeable par rapport au coût lié au respect des rectangles de navigation et à l'écart des biefs envers leur NNL.

Afin de permettre l'optimisation simultanée de plusieurs biefs de tailles variables, le coût lié à l'écart des biefs avec leur NNL respectif est normalisé en divisant par le coût maximum possible de ce bief $(NNL_e - w_e)^2$. Cette normalisation est requise à cause de la différence significative de la taille des biefs. Par exemple, sur le réseau des Hauts-de-France, le bief Fontinettes-Flandres présente un écart de volume entre le NNL et le HNL de 1 783 m^3, tandis que le même écart sur le bief Cuinchy-Fontinettes est de 109 980 m^3. Sans normalisation, une diminution de 1 000 m^3 sur les biefs Cuinchy-Fontinettes et Fontinettes-Flandres serait pénalisée de la même façon. Or, cela correspond à un écart de niveau de près de 3 centimètres pour Fontinettes-Flandres et de moins de 1 millimètres pour Cuinchy-Fontinettes.

Le modèle et la discrétisation des états et actions ayant été décrits pour la gestion de la ressource en eau sur les voies navigables, il est maintenant possible de mettre en place une approche d'optimisation stochastique.

3.5 Applicabilité de l'état de l'art

Les approches existantes dans la littérature et présentées dans le chapitre précédent pour permettre la modélisation et l'optimisation de grands systèmes ne sont pas forcément pleinement applicables à la classe de problèmes ici présentée. Leur applicabilité sur cette classe de problèmes est donc discutée.

3.5.1 MDP

Les processus décisionnels markoviens permettent la modélisation de la gestion de la ressource en eau des voies navigables, comme décrit dans la section précédente. Néanmoins, une telle modélisation ne permet pas le passage à l'échelle. Pour rappel (voir section 2.1.3 page 28), une résolution optimale implique de visiter au moins une fois chaque transition du modèle à chaque itération. Considérons un réseau de huit composants chacun divisé en neuf intervalles. Les composants sont interconnectés grâce à sept effecteurs composés de six intervalles. Même si l'optimisation se déroule sur un seul pas de temps ($\tau = 1$), il sera difficile de stocker la fonction de transition. En effet, le MDP sera composé de $9^8 \times (\tau + 1)$ états et de 6^7 actions, d'où :

$$|T| = |S| \times |A| \times |S| = (9^8 \times 2)^2 \times 6^7 \simeq 2{,}0 \times 10^{21} \tag{3.33}$$

En utilisant les propriétés du temps, c'est-à-dire que le temps augmente de 1 après chaque action, il est possible de simplifier la fonction de transition en ne gardant que les transitions temporellement possibles. Seules les transitions $T(s,a,s')$ respectant $t_{s'} = t_s + 1$ seront stockées, où t_s est le pas de temps de l'état s. Ainsi, au maximum :

$$|T| = \frac{1}{\tau + 1} \times |S| \times |A| \times |S| = 2 * (9^8)^2 \times 6^7 \simeq 1{,}0 \times 10^{21} \tag{3.34}$$

Bien que des optimisations puissent être faites pour réduire la taille de la fonction de transition, celle-ci reste très importante. Prenons l'exemple où la fonction de transition doit être stockée en mémoire. En supposant qu'il soit possible de stocker chaque valeur de la fonction en utilisant 1 octet, il serait nécessaire d'utiliser approximativement 10^{21} octets, soit 10^{12} gigaoctets de mémoire. Une possibilité pour réduire l'espace mémoire serait de calculer les valeurs de la fonction à la volée. Cependant bien que l'espace mémoire requis soit grandement diminué, le temps de calcul deviendrait plus important, avec un ajout d'un minimum de 10^{21} opérations par itération pour l'obtention des transitions.

3.5.2 MDP factorisé

Aux premiers abords, ce type de modèle semble se prêter favorablement à l'utilisation de MDPs factorisés. Les états du système peuvent être représentés par un ensemble de variables représentant l'évolution de la ressource de chaque composant. De même, une action peut être définie comme un ensemble de variables, chacune correspondant à la quantité déplacée par un des effecteurs.

Cependant, bien que l'évolution des variables d'états, représentant les composants, ne dépende que de leurs valeurs précédentes et du résultat des effecteurs, il n'est pas possible d'utiliser cette propriété pour factoriser le modèle. En effet, certaines incertitudes sur les effecteurs ont le même impact sur la quantité de ressource des composants sources et des composants cibles.

Cette caractéristique empêche la modélisation de l'évolution globale du système comme une combinaison d'évolutions locales.

Prenons le cas d'un effecteur qui a 50% de chance de sur-déplacer la ressource d'un composant source i vers un composant cible j. Ainsi une action peut résulter de façon équiprobable en le déplacement d'une ou de deux unités de ressource. Par définition du problème, sur ce pas de temps, deux configurations peuvent se produire :

— le composant i perd une unité de ressource et le composant j en gagne une (50%) ;
— le composant i perd deux unités de ressource et le composant j en gagne deux (50%).

Si l'évolution de chaque composant est calculée de façon séparée, comme dans une potentielle modélisation factorisée, il sera considéré que le composant i perdra une ou deux unités de ressource et que le composant j gagnera une ou deux unités de ressource. De ce fait, sur ce pas de temps quatre configurations, dont deux incorrectes, pourraient se produire :

— le composant i perd une unité de ressource et le composant j en gagne une (25%) ;
— le composant i perd deux unités de ressource et le composant j en gagne deux (25%) ;
— le composant i perd une unité de ressource et le composant j en gagne deux (25%) ;
— le composant i perd deux unités de ressource et le composant j en gagne une (25%).

Ces configurations montrent l'inconsistance de la prise en compte séparée, par les composants, des incertitudes des effecteurs.

De plus, l'évolution d'un composant dépend de l'ensemble des effecteurs qui l'affectent. La prise en compte d'un effecteur implique d'observer son impact sur l'ensemble des composants qu'il connecte. De ce fait, une factorisation efficace n'est pas possible sans impliquer une approximation importante des incertitudes.

3.5.3 MDP décomposé

Les modèles considérés par la classe de problèmes ne sont pas particulièrement adaptés aux méthodes de décomposition. En effet, la décomposition en parallèle consiste à représenter un problème par un produit de MDPs locaux. Une solution intuitive de décomposition est de séparer le problème en gestion de chaque composant. Or, les effecteurs affectent fortement plusieurs composants en les reliant les uns aux autres. De ce fait, la décomposition en MDPs locaux n'est pas applicable.

Les décompositions en série se basent sur l'existence de partitions faiblement connectées de l'espace d'état. Or, un état du système correspond à une capture du réseau complet ce qui fait qu'il n'existe pas de décomposition évidente en problèmes locaux. Une décomposition intuitive serait de diviser le problème en fonction des quantités de ressources des composants, et ainsi avoir des plans de gestion pour les différentes conditions de fonctionnement, par exemple :

— les composants ont tous trop peu de ressource ;
— les composants ont tous trop de ressource ;
— les composants sont dans un état optimal ;

— ....

Or ce genre de décompositions reviendrait simplement à utiliser une décomposition en intervalles de plus grandes tailles, dans le cadre d'une résolution hiérarchisée, ce qui ne présente qu'un intérêt limité. Ici les problèmes de passage à l'échelle sont d'avantage liés aux nombres de composants et d'effecteurs qu'à leurs nombres d'intervalles.

3.5.4 MDP Monte-Carlo

Les méthodes de planification basées sur Monte-Carlo peuvent s'appliquer à n'importe quel type de problème, si un simulateur est accessible. Elles permettent ainsi de déterminer une action à effectuer en un temps fini. Cependant, pour cela une exploration partielle des évolutions possibles du problème est réalisée. Cette exploration se focalise sur les états probables et intéressants qui peuvent être atteints en partant de l'état initial. La solution produite est par essence définie sur le sous-ensemble des états explorés.

Or pour cette thèse, les problèmes considérés sont très affectés par des incertitudes. De ce fait, il est possible que des événements rares ne soient pas pris en compte et affectent négativement la solution, notamment lorsque le nombre d'actions est important. De plus, il est nécessaire de calculer à chaque pas de temps quelle action doit être effectuée, ce qui peut nuire à la résilience lorsqu'un événement imprévu se produit et que l'obtention d'une solution de qualité n'est pas assez rapide, à cause de la taille du problème traité. Ainsi, il est considéré que les approches basées Monte-Carlo ne sont pas adaptées pour les problèmes considérés par cette thèse.

3.5.5 Dec-SIMDP

Les Dec-SIMDPs exploitent la localité des interactions entre les agents, lorsque celles-ci sont limitées sur un faible nombre d'états, dans un environnement partagé. Ainsi, un agent serait défini par un ensemble d'effecteurs, et aurait comme espace d'état les états joints des composants qu'ils affectent. Pour la classe de problèmes considérée, les interactions entre des groupes d'effecteurs sont extrêmement localisées. Deux effecteurs n'étant en interaction que s'ils affectent un même composant. Dans ce cas, si deux agents sont en interaction, alors au vu de la définition des états, l'espace d'états ne formerait qu'une seule zone d'interaction. Ainsi aucune réduction du nombre d'états ne serait possible. Cette approche étant définie pour des problèmes avec un faible nombre d'interactions, elle ne possède pas d'intérêt ici.

3.5.6 ND-POMDP

Les ND-POMDPs nécessitent des caractéristiques spécifiques de modélisation qui ne sont pas compatibles avec la classe de problèmes considérée. Cette modélisation se base sur l'indépendance des transitions des actions définies pour chaque agent avec un besoin de coordination pour maximiser les gains de leur fonction de récompense. Or, pour les problèmes considérés, les composants sont connectés les uns aux autres et cela implique une série de dépendances localisées. Ceci ne

permet pas d'avoir l'indépendance des fonctions de transition des agents et ainsi d'utiliser une modélisation/résolution basée sur les ND-POMDPs.

3.6 Conclusion

Ce chapitre définit une classe de problèmes de gestion prédictive sous incertitudes d'une ressource partagée sur un réseau. Ce type de problème est défini par des composants, stockant cette ressource, reliés entre eux et à leur environnement par des effecteurs. Ces effecteurs permettent le déplacement de la ressource sur le réseau. Le but est de maintenir les objectifs de gestion de chaque composant.

Une définition formelle des états et des actions d'un MDP décrivant ces problèmes a été proposée. Elle implique la discrétisation des valeurs stockables par chaque composant et des valeurs déplaçables par chaque effecteur. À partir de cette discrétisation, une définition de la fonction de coût est possible. Les deux types d'incertitudes affectant les problèmes ont été définis ainsi que leur prise en compte dans la fonction de transition. Cette modélisation a été brièvement illustrée sur un problème de gestion de l'eau sur les voies navigables.

À notre connaissance, les approches de la littérature ne permettent pas de traiter la classe de problèmes considérée tout en garantissant un certain degré de résilience. Ces approches sont inadaptées à la problématique à cause d'une incapacité à passer à l'échelle, de l'exploitation de caractéristiques du modèle non présentes ou encore par l'abstraction de certains états *a priori* peu intéressants qui peut nuire à la résilience. Les approches de décomposition en parallèle et par ND-POMDP semblent les plus intéressantes pour notre problématique malgré leur contraintes d'indépendance de l'évolution de différentes sous-parties du problème. Ainsi une nouvelle approche exploitant les propriétés de la classe de problèmes définie est proposée, en s'inspirant notamment du traitement des ND-POMDPs avec l'algorithme LID-JESP, afin de répondre à la problématique définie. Cette approche est décrite dans le chapitre suivant.

Chapitre 4

Coordination hors-ligne de planifications locales

De nombreuses méthodes ont été proposées dans la littérature afin de permettre l'optimisation sous incertitudes de réseaux complexes de tailles importantes. Cependant, à notre connaissance et comme présenté dans le chapitre précédent, il n'existe pas d'outils de modélisation et de résolution permettant un passage à l'échelle pour les systèmes correspondant à la classe de problèmes qui a été définie. Celle-ci représente des problèmes constitués de composants naturellement distribués sur un réseau partageant une ressource quantitative dont la gestion doit être optimisée. Pour y répondre, une nouvelle méthodologie est donc nécessaire. La méthode proposée reprend les travaux présentés dans [Desquesnes et al., 2017c] et [Desquesnes et al., 2017b].

La méthode proposée est multi-agent et repose sur une définition locale des agents. Cette localité des agents rend nécessaire de les coordonner afin d'optimiser le système global. Une décomposition automatique d'un problème en agents est donc nécessaire afin de permettre d'obtenir un partitionnement efficace du modèle. Un algorithme local à chaque agent est ainsi défini. Il décrit les étapes que chaque agent doit effectuer pour résoudre son problème local et ainsi obtenir sa politique locale. Un agent pourra utiliser cette politique sans avoir besoin de communiquer avec les autres agents durant son application.

Un exemple d'exécution de l'algorithme est proposé en utilisant une version simplifiée de l'algorithme introduit. Cet exemple se base sur un problème de satisfaction de contraintes afin de clarifier le fonctionnement de l'algorithme. Une comparaison avec une méthode optimale existante est réalisée afin d'observer les avantages et les limitations de l'approche proposée.

4.1 Présentation de l'algorithme OCLP

L'approche proposée, nommée « coordination hors-ligne de planifications locales » (OCLP - Offline Coordination of Local Plannings), s'inspire du Distributed Breakout Algorithm (DBA) [Yo-

koo and Hirayama, 1996] et de son application aux ND-POMDPs [Nair et al., 2005]. La problématique des réseaux de composants traitée est en effet semblable aux problèmes de contraintes traités par le DBA, où des valeurs doivent être choisies afin de maximiser le respect des contraintes sur ces valeurs. La notion d'optimisation de ce type de problèmes est introduite par algorithme LID-JESP pour les ND-POMDPs. Contrairement à la classe de problèmes considérée, l'environnement d'un ND-POMDP peut être divisé en plusieurs parties dont les évolutions sont totalement indépendantes.

L'approche OCLP consiste à distribuer sur plusieurs agents l'optimisation des capacités de contrôle du réseau modélisé. Chaque agent possède un modèle local correspondant aux composants qu'il affecte grâce à ses effecteurs. Ainsi, contrairement à l'algorithme LID-JESP, les agents n'auront pas une connaissance complète de l'environnement avec lequel ils sont en interaction. Il optimisera sa gestion localement, mais en se coordonnant avec les autres agents. Ceci conduit à l'obtention pour chaque agent d'une politique locale de gestion, de ses effecteurs, qui permet une gestion résiliente du réseau concerné.

4.1.1 Distribution en agents

La distribution d'un réseau sur plusieurs agents locaux, ne s'occupant que d'une fraction du système modélisé, facilite le passage à l'échelle. Cela permet le traitement de problèmes réels. Similairement au DBA, un agent est défini par les effecteurs qu'il contrôle (les variables à assigner). Ainsi, un agent n'est pas défini pour un composant, mais par l'ensemble des composants du réseau qu'il affecte avec ses effecteurs. Chaque agent a pour but d'optimiser localement la gestion de la ressource des composants affectés par ses effecteurs. Un agent aura alors uniquement connaissance de ces composants, à l'inverse de l'algorithme LID-JESP où chaque agent aura une connaissance du problème dans son ensemble. Les composants étant le plus souvent dépendants de plusieurs effecteurs, les actions d'un agent impacteront vraisemblablement les états d'autres agents. Un exemple est visible sur la figure 4.1, où quatre agents (α, β, γ et δ) se répartissent les effecteurs. Trois des agents possèdent la vision de plusieurs composants ; seul l'agent γ ne peut observer qu'un seul composant. Dans cet exemple, le composant 1 est affecté par les actionneurs 0, 1, 2 et 3 qui sont contrôlés par deux agents différents (α et β). De ce fait, les agents devront se coordonner pour permettre une optimisation efficace des quatre composants qu'ils affectent.

4.1.2 Description de l'algorithme

L'algorithme OCLP se base donc sur une décomposition du modèle en agents. Dans le but d'optimiser la gestion du système global, les agents doivent échanger leur politique locale avec les autres agents. Cet échange est nécessaire afin que les agents puissent coordonner leurs politiques locales et ainsi éviter des actions contradictoires sur un même composant.

Dans un grand nombre d'applications réelles, pour une décomposition donnée, un agent n'aura pas besoin de communiquer avec l'ensemble des autres agents pour calculer sa politique locale

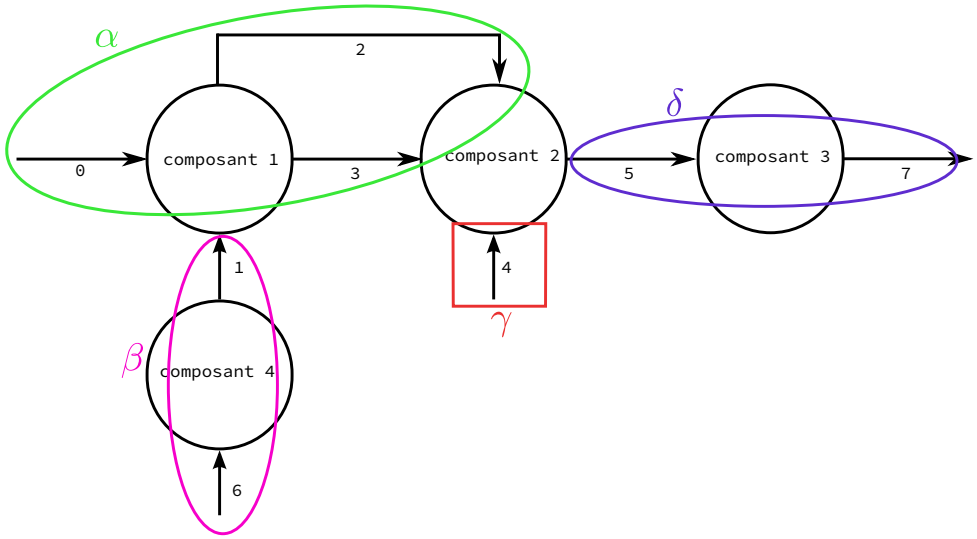

FIGURE 4.1 – Exemple de décomposition en agents, les composants sont représentés par des cercles, les actionneurs par des arcs et les agents sont les ensembles colorés d'actionneurs

de gestion. En effet, pour optimiser la gestion de quelques composants qu'il affecte, un agent n'a besoin que de connaître la politique des agents qui affectent ses composants. Deux agents affectant un même composant seront considérés comme voisins et devront communiquer pour planifier efficacement la gestion de leurs composants communs. Un voisinage est défini comme un ensemble composé d'un agent et de tous ses voisins.

Pour permettre une bonne coordination des agents, un calcul itératif des politiques locales en utilisant le principe de coévolution est utilisé. La coévolution est un principe biologique où pour deux espèces en symbiose, l'évolution d'une espèce entraînera l'évolution de la seconde [Ehrlich and Raven, 1964]. Pour la planification et la classe de problèmes considérée, la coévolution consiste à ne permettre qu'un seul changement de politique par voisinage et par itération. Ce changement peut avoir comme effet la dégradation de la politique locale d'un voisin, ou l'ouverture de nouvelles possibilités de gestion. Cela pourrait entraîner la modification de la politique de ce voisin dans une itération ultérieure. Interdire aux voisins de changer simultanément leur politique permet de limiter les changements conflictuels. Par exemple, pour un réseau de voies navigables, si un bief est en surplus d'une certaine quantité d'eau par rapport au NNL et si deux agents affectant ce bief décident simultanément de modifier leur politique afin de déplacer ce volume d'eau, alors le bief pourrait se retrouver en manque d'eau.

Basé sur ce principe, l'algorithme 4 propose un protocole de communication permettant d'obtenir une solution potentiellement sous-optimale, mais efficace, pour la classe de problèmes considérée en utilisant la modélisation en MDP présentée dans la section 3.2. Pour cet algorithme, les agents et les communications peuvent être considérés comme synchronisés, car chaque itération se base sur les politiques et les gains de l'itération précédente pour les agents du voisinage.

Algorithme 4 Coordination hors-ligne de planifications locales pour un agent α

1: Création d'une politique locale initiale $\pi_{\alpha_0} : S_\alpha \rightarrow A_\alpha$
2: $d \leftarrow$ distance maximale entre deux agents
3: $counter \leftarrow d$
4: $it \leftarrow 0$
5: **répéter**
6: Adapter et échanger $\pi_{\alpha_{it}}$ avec chaque voisin (voir équation 4.1)
7: Mise à jour de MDP_α^{it} en MDP_α^{it+1} en utilisant les politiques reçues
8: Calcul de π'_α la politique optimale de MDP_α^{it+1}
9: $g_\alpha^{it} \leftarrow gain(\pi'_\alpha, \pi_\alpha^{it})$ de MDP_α^{it+1}
10: Échanger g_α^{it} de chaque voisin
11: $G_\alpha^{it} \leftarrow$ l'ensemble des gains du voisinage
12: $\pi_\alpha^{it+1} \leftarrow \pi'_\alpha$ si $g_\alpha^{it} = \max(G_\alpha^{it})$ sinon π_α^{it}
13: $counter \leftarrow d$ si $g_\alpha^{it} > 0$ sinon $counter - 1$
14: Échanger $counter$ avec chaque voisin
15: $Counters^{it} \leftarrow$ l'ensemble des compteurs du voisinage
16: $counter \leftarrow \max(Counters^{it})$
17: $it \leftarrow it + 1$
18: **jusqu'à** $counter = 0$

À chaque itération it, chaque agent échange sa politique actuelle avec ses voisins (ligne 6). Les agents n'ont pas forcément le même ensemble d'états, mais possèdent par définition un sous-ensemble de composants et donc de sous-états en commun. De ce fait, contrairement à l'algorithme LID-JESP, la politique échangée devra être adaptée à la connaissance commune du réseau partagée par les deux agents. Cela signifie que la politique transmise sera définie uniquement sur les composants que les deux agents affectent. La politique d'un agent α adaptée à la connaissance d'un agent β, noté π_β^α, correspond à une politique probabiliste sur les états communs de α et β (équation 4.1). Il s'agit d'une politique, où une distribution de probabilité sur les actions que pourra faire l'agent α est associé à chaque état partagé par les agents α et β, c'est-à-dire une assignation des composants qu'ils affectent tous les deux.

$$\pi_\beta^\alpha(s_{\alpha\beta}) = \{(\pi_\alpha(s_\alpha),\, p(s_\alpha|s_{\alpha\beta})),\, \forall s_\alpha \in S_\alpha\} \tag{4.1}$$

où s_α est un état du modèle de l'agent α et $s_{\alpha\beta}$ est une assignation des états locaux correspondant aux composants partagés par les agents α et β.

Lorsque l'agent α a reçu les politiques adaptées de tous ses voisins, il peut mettre à jour son modèle actuel $\text{MDP}_\alpha^{it} = \langle S_\alpha, A_\alpha, T_\alpha^{it}, R_\alpha \rangle$ en MDP_α^{it+1}. De cette façon, l'agent prend en compte une estimation de l'impact qu'auront les autres agents sur les composants qu'il affecte (ligne 7). Ceci est fait en redéfinissant la fonction de transition de l'agent en fonction des distributions de probabilités sur les actions des politiques obtenues, voir l'équation 4.2. Cette politique adaptée

est une estimation puisqu'elle ne prend pas en compte les changements de politiques pouvant avoir eu lieu ailleurs dans le système lors de la même itération.

$$\begin{aligned} T_{\alpha}^{it+1}(s_{\alpha},a_{\alpha},s'_{\alpha}) &= \sum_{j \in jointactions(s_{\alpha})} P(s'_{\alpha}|s_{\alpha},a_{\alpha},j) \times P(j) \\ jointactions(s_{\alpha}) &= \prod_{\beta \in N_{\alpha}} \pi_{\alpha}^{\beta}(s_{\alpha\beta}),\ s_{\alpha\beta} \subseteq s_{\alpha} \end{aligned} \tag{4.2}$$

où N_{α} représente l'ensemble des agents voisins de l'agent α.

En utilisant son nouveau modèle MDP^{it+1}, chaque agent calcule une politique optimale π' pour ce nouveau modèle tenant compte des changements des voisins (ligne 8). Comme indiqué précédemment, pour éviter les changements conflictuels, au plus un agent par voisinage aura la possibilité de garder sa nouvelle politique lors d'une itération.

L'amélioration de la politique nouvellement calculée (π'_{α}) par rapport à la politique actuelle de l'agent α (π_{α}^{it}) est utilisée pour déterminer quels agents auront la possibilité de garder leur nouvelle politique. Cette amélioration est obtenue grâce à une fonction heuristique (ligne 9) :

$$gain : (S \to A) \times (S \to A) \to \mathbb{R}^{+} \tag{4.3}$$

La fonction heuristique a pour but de guider l'exploration pour l'optimisation du problème et de permettre une détection locale de la convergence de l'algorithme. Une heuristique simple et générique consiste à faire la différence entre les fonctions de valeur de deux politiques sur le dernier modèle de l'agent.

À chaque itération, les agents échangent leur gain avec leur voisinage, seul l'agent ayant la plus grande amélioration non nulle de son voisinage peut garder sa nouvelle politique (ligne 12). Les autres agents conservent leur ancienne politique. Un exemple de détermination des agents ayant le plus grand gain de leur voisinage est visible sur la figure 4.2. Dans l'exemple, un agent, représenté par un cercle, possède une amélioration de 7 et reçoit les valeurs de ses deux voisins : 5 et 3. Ayant la plus grande valeur de son voisinage, l'agent pourra garder sa nouvelle politique à cette itération (double cercle coloré). En cas d'égalité entre plusieurs agents dans un même voisinage, un ordonnancement sur les agents est utilisé. Il est possible d'utiliser un ordre lexicographique pour déterminer quel agent garde la politique : l'agent 1 est prioritaire sur l'agent 2.

Afin d'éviter que seul un agent au total ne puisse mettre à jour sa politique par itération, il serait possible d'ignorer les gains des agents qui sont bloqués par un autre voisinage. C'est-à-dire, si un agent garde sa politique, il inhiberait les gains de tous ses voisins. Un exemple est visible sur la figure 4.3 : l'agent avec l'amélioration maximale (9) empêche son voisin direct de changer de politique. Les voisins de ce voisin ont des gains inférieurs et ne pourront donc pas changer leur politique. Cette succession décroissante de gains ne permet donc qu'à un seul agent de garder son plan. Dans ce cas, puisque l'agent dont le gain est de 9 garde sa politique, son voisin avec un gain de 6 pourrait être ignoré. L'agent avec un gain de 4 deviendrait donc le maximum de son

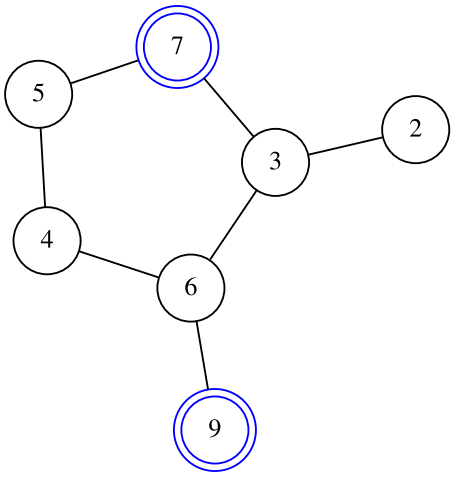

FIGURE 4.2 – Exemple de détermination du plus grand gain. Chaque cercle est un agent et la valeur en son centre son gain, les arêtes indiquent les relations de voisinage et le double cercle coloré indique les agents ayant la plus grande amélioration de leur voisinage

voisinage et pourrait garder sa politique ; de même pour l'agent avec un gain de 2 (figure 4.4). Cette approche nécessiterait cependant une plus grande utilisation des communications entre agents.

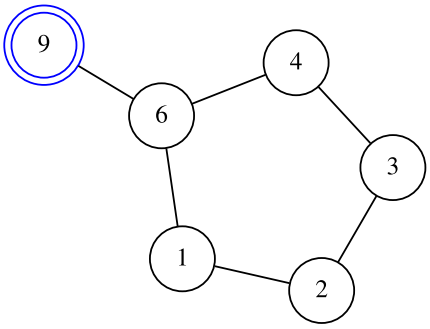

FIGURE 4.3 – Exemple où seul un agent peut changer de politique

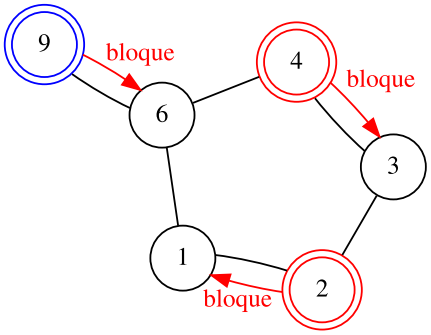

FIGURE 4.4 – Ignorer les agents bloqués permet d'augmenter le nombre de politiques conservées

Pour garantir que tous les agents s'arrêtent au même moment si et seulement si aucun agent ne peut plus améliorer sa politique locale, des compteurs sont utilisés par chaque agent. Le compteur d'un agent a pour but d'estimer localement le nombre d'itérations restantes avant la convergence.

Chaque agent initialise son compteur à une valeur commune $d > 0$ (ligne 3). La valeur de d est

assignée une fois pour toutes lors de l'initialisation de la résolution par un agent externe possédant des informations sur l'ensemble du réseau. Pour garantir la terminaison simultanée des agents, d prendra comme valeur la distance maximale entre deux agents sur le système (équation 4.5). À la fin d'une itération, si un agent a obtenu une politique avec une amélioration non nulle (gardée ou non), alors le compteur est réinitialisé à d ; dans les autres cas le compteur est réduit de 1 (ligne 13). Ensuite, chaque agent échange son compteur avec ses voisins et ne garde que le plus grand du voisinage (ligne 16). Lorsque le compteur d'un agent atteint 0, celui-ci arrête sa résolution et sa politique locale actuelle sera celle qu'il utilisera à l'exécution. L'algorithme est garanti de s'arrêter après au plus :

$$d = \max_{\alpha,\beta \in Ag^2} distance(\alpha,\beta) \tag{4.4}$$

itérations si et seulement si tous les agents sont dans un optimum local ; une preuve proposée dans [Nair et al., 2005] est applicable ici et présentée ci-après.

$$distance(\alpha,\beta) = \begin{cases} 1 & \text{si } \beta \in N_\alpha \\ 1 + \min\limits_{\gamma \in N_\alpha} distance(\gamma,\beta) & sinon \end{cases} \tag{4.5}$$

Cependant, similairement à l'algorithme DBA [Yokoo and Hirayama, 1996], l'algorithme proposé n'offre pas de garantie de convergence vers une solution localement optimale. De ce fait, l'arrêt de l'algorithme ne peut être garanti. Néanmoins, lorsqu'une solution localement optimale est atteinte, l'algorithme terminera en un temps fini.

4.1.3 Preuve de terminaison

Une preuve par l'absurde est utilisée pour montrer, que si un agent arrête sa résolution lorsqu'il détecte localement la convergence grâce à son compteur, alors tous les autres agents ont localement détecté la convergence.

Supposons qu'un agent α ne commence pas l'itération it, car son compteur a atteint 0 et que par conséquent sa résolution s'est terminée à l'itération $it-1$, mais que d'autres agents aient un compteur strictement positif. Cela implique qu'à l'itération $(it-d)$ il existe au moins un agent β qui peut améliorer sa politique et ainsi réinitialiser son compteur ($counter_\beta(it-d) = d$). Les compteurs étant décrémentés d'au plus 1 à chaque itération puis étant propagés à travers le voisinage, alors à l'itération $(it-d+distance(\alpha,\beta))$ la borne minimale du compteur de l'agent α peut être définie par $d - distance(\alpha,\beta)$. De ce fait, le compteur de l'agent α à l'itération $(it-1)$ peut être borné par :

$$\begin{aligned} counter_\alpha(it-1) &\geq d - distance(\alpha,\beta) + (-1) \times (d - distance(\alpha,\beta) - 1) \\ &= 1 \end{aligned} \tag{4.6}$$

Comme $counter_\alpha(it-1) \geq 1$, l'agent α devra effectuer l'itération it en contradiction avec l'hypothèse initiale. Par conséquent, si l'algorithme se termine pour un agent, alors tous les agents ont un compteur à 0.

À l'inverse, si tous les agents ont atteint un optimum local alors, les gains de chaque agent resteront nuls et les compteurs ne sont jamais réinitialisés à d. Avec les échanges de compteurs, la valeur du plus grand compteur diminuera de 1 à chaque itération. Donc après d itérations tous les compteurs auront atteint 0 et les agents s'arrêteront.

4.1.4 Cycles

Similairement au DBA [Yokoo and Hirayama, 1996], l'algorithme OCLP proposé n'offre aucune garantie quant à la convergence vers une solution. Dans le cas de l'algorithme OCLP, les agents se sont retrouvés coincés dans un cycle de mise à jour de politique, lors du calcul de la politique jointe d'un scénario particulier de gestion d'un réseau réel (section 5.3 page 113).

Lorsqu'un agent α change sa politique de π_α^i à π_α^j, cela affectera son voisin β. Ce dernier, pour s'adapter à ce changement dans son voisinage, mettra à jour son plan. Ce changement sera pris en compte par l'agent α en changeant son plan de π_α^j en π_α^i. Ainsi les agents α et β se retrouvent dans un cycle de résolution de longueur 2, où une suite de politiques sera répétée sur une période potentiellement infinie, empêchant ainsi les compteurs d'atteindre 0.

Dans un grand nombre de cas, ces cycles seront brisés automatiquement lorsque des changements de politiques d'autres agents, plus loin dans le réseau, leur seront transmis. Cependant, dans d'autres cas, les autres agents ont atteint des optimums locaux ou leurs modifications n'affectent pas le cycle. Dans ces circonstances, les cycles doivent être détectés et brisés automatiquement afin de permettre à l'algorithme de continuer et de s'arrêter.

Il existe plusieurs méthodes pour permettre la détection des cycles lors de la résolution ; notamment, en observant l'évolution des gains des voisinages au cours des itérations. Lorsqu'une séquence de valeurs de gains, ou de politiques locales, successives identiques à une séquence précédente est détectée cela signifie que le voisinage se trouve vraisemblablement dans un cycle. La méthode employée pour briser les cycles est de ne plus garder les politiques et de décrémenter les gains jusqu'à ce qu'une modification potentielle arrive des autres voisinages. Puisqu'il est nécessaire d'attendre au minimum qu'un cycle complet soit réalisé afin de le détecter, cela peut avoir un impact considérable sur le temps de calcul.

4.2 Exemple d'utilisation de l'algorithme basé sur un problème d'optimisation de contraintes

L'algorithme OCLP a été introduit pour résoudre des problèmes d'optimisation de partage de ressource entre plusieurs composants en utilisant des actions distribuées. Un exemple illustratif est présenté sur la figure 4.5. Il s'agit d'un réseau d'entrepôts partageant une même ressource, des objets identiques. Les entrepôts, notés de 0 à 4, sont schématisés par des carrés.

Chaque entrepôt a une évaluation différente quant à la quantité d'objets stockés. L'équation 4.7 spécifie le coût lié au stockage pour chaque entrepôt. s_i^* représente le nombre optimal

d'objets à stocker dans le $i^{\text{ème}}$ entrepôt et C_i est une fonction pénalisant, de façon spécifique à l'entrepôt, l'écart à cette quantité optimale. Chaque entrepôt possède une capacité minimale de 0, une capacité initiale correspondant à l'état de l'entrepôt au début de la gestion et une capacité maximale qui est spécifiée dans chaque carré par « initiale/maximale ». Il est uniquement possible de déplacer les objets d'entrepôt en entrepôt, dans le sens défini par les flèches (o, p, q et r) tout en respectant les capacités de transfert définies par les intervalles associés aux flèches. Les actions sont considérées comme instantanées et simultanées. L'entrepôt 1 est un cas particulier puisqu'il ne possède pas de quantité optimale d'objets à stocker. Le respect des contraintes de capacités (minimale et maximale) et le seul objectif de ce composant.

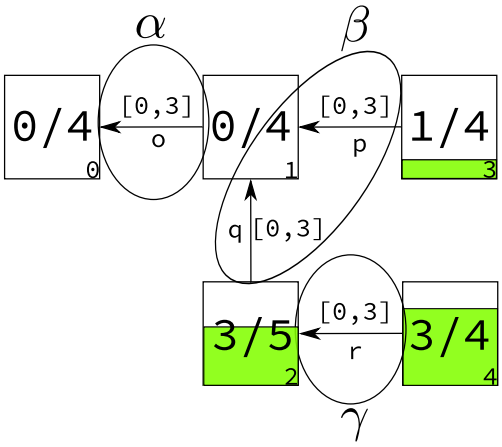

FIGURE 4.5 – Réseau d'entrepôts composé de 5 composants (carrés), 4 actionneurs (arcs) et 3 agents (α, β et γ) représenté dans son état initial

$$\begin{array}{llllll} s_0^* = 4 & s_1^* = \emptyset & s_2^* = 3 & s_3^* = 0 & s_4^* = 0 \\ C_0(s_0) = & |4 - s_0| & & C_3(s_3) = & |0 - s_3| \\ C_1(s_1) = & 0 & & C_4(s_4) = & (0 - s_4)^2 \\ C_2(s_2) = & |3 - s_2| & & & \end{array} \tag{4.7}$$

Trois agents, α, β, γ, vont gérer les transferts entre les différents entrepôts qu'ils contrôlent. L'agent α contrôle le transfert $\{o\}$. Cela lui donne une vision de l'état des entrepôts $\{0, 1\}$. Les agents β et γ contrôlent respectivement $\{p, q\}$ et $\{r\}$. Ainsi, ils peuvent connaître respectivement l'état des entrepôts $\{1, 2, 3\}$ et $\{2, 4\}$. L'agent α a pour seul voisin l'agent β. Ils partagent la gestion de l'entrepôt 1. Similairement, l'agent γ a pour seul voisin β grâce au partage de l'entrepôt 2. Par transitivité du voisinage, β a deux voisins α et γ lui permettant de connaître tous les agents. Néanmoins, cela n'implique pas une connaissance complète du système, il ne voit que les entrepôts qu'il affecte : $\{1, 2, 3\}$. Le chemin maximal entre deux agents dans ce système est de taille $d = 2$.

Pour un agent, déplacer un objet d'un entrepôt à un autre nécessite que l'entrepôt source ne soit pas vide et que l'entrepôt cible ne soit pas plein. Dans le cas contraire, les objets excédentaires sont détruits et les objets manquants sont achetés, induisant un coût rédhibitoire de $+\infty$. Chaque agent a donc un simple problème sous contraintes à optimiser, visant à minimiser la somme des distances de chaque agent à sa valeur optimale comme indiqué dans l'équation 4.7. Ce problème d'optimisation est résolu à l'aide de l'algorithme OCLP, en adaptant les caractéristiques liées à la modélisation par MDPs, dans un souhait de simplification de l'exemple.

Le problème considéré consiste à optimiser l'ensemble des entrepôts sur un horizon 1, sachant que l'état initial du système est $\{0,\ 0,\ 3,\ 1,\ 3\}$. Pour ce scénario, le résultat des actions est déterministe et chaque agent choisit son affectation initiale (ligne 1 de l'algorithme 4) en supposant que ses voisins n'existent pas. Ce qui donne les affectations initiales suivantes :

$$\alpha\ :\ (o=0) \qquad \beta\ :\ (p=1,\ q=0) \qquad \gamma\ :\ (r=2) \tag{4.8}$$

L'application de ces affectations initiales amène le système dans l'état suivant :

$$\{0,1,5,0,1\} \text{ avec comme valeur globale :} \tag{4.9}$$

$$C^0_\omega = C_0(0) + C_1(1) + C_2(5) + C_3(0) + C_4(1) = 4 + 0 + 2 + 0 + 1 = 7 \tag{4.10}$$

où le coût global de l'affectation C^{it}_ω est la somme des coûts de chaque agent après l'exécution de la $it^{\text{ème}}$ affectation, voir la relation 4.7.

Dans cet exemple, les affectations des agents donnent les actions qui seront faites, quel que soit l'état du système. L'adaptation pour le voisinage consiste simplement à n'envoyer que les actions qui affectent les composants partagés. Les compteurs des agents sont initialisés à $d = 2$ (ligne 3).

$$counter_\alpha\ =\ 2 \qquad counter_\beta\ =\ 2 \qquad counter_\gamma\ =\ 2 \tag{4.11}$$

Au début de la première itération, les agents commencent par échanger leurs affectations (ligne 6). L'agent α envoie son action $(o = 0)$ à l'agent β. Similairement, γ envoie $(r = 2)$ à β. Ce dernier communique $(p = 1,\ q = 0)$ à l'agent α et $(q = 0)$ à l'agent γ. Ainsi, chaque agent peut modifier sa vision de l'environnement pour prendre en compte les choix annoncés de leurs voisins. Après un nouveau calcul d'affectation (ligne 8), les affectations des agents sont :

$$\alpha\ :\ (o=1) \qquad \beta\ :\ (p=1,\ q=2) \qquad \gamma\ :\ (r=2) \tag{4.12}$$

Ce qui conduit aux gains suivants pour les agents :

$$g_\alpha = 1 \qquad \text{et} \qquad g_\beta = 2 \qquad \text{et} \qquad g_\gamma = 0 \tag{4.13}$$

ici, les gains sont calculés en comparant l'évaluation du modèle connu par l'agent en appliquant la nouvelle affectation qu'il vient de calculer et son affectation courante. Les affectations du voisinage, considérées pour le calcul, sont celles reçues au début de l'itération.

Les gains des agents sont comparés par voisinage pour déterminer quel agent peut garder l'affectation qu'il vient de calculer (ligne 12). Le gain de γ est nul, ce qui signifie qu'il possède déjà une affectation localement optimale pour le modèle actuel. Ainsi, aucun changement ne sera donc appliqué à cet agent durant cette itération. L'agent β possède le gain maximum de son voisinage (2) ce qui implique qu'il gardera l'affectation qu'il vient de calculer. Pour α, un agent de son voisinage garde déjà son affectation (β), il ne pourra donc pas garder sa nouvelle affectation. La nouvelle affectation jointe résultant de cette itération est :

$$\alpha \ : \ (o = 0) \qquad \beta \ : \ (p = 1, \, q = 2) \qquad \gamma \ : \ (r = 2) \tag{4.14}$$

L'application de cette affectation amène dans l'état suivant :

$$\{0, \, 3, \, 3, \, 0, \, 1\} \text{ avec comme valeur globale } C^1_\omega = 5. \tag{4.15}$$

Les agents ayant trouvé une affectation meilleure que leur affectation actuelle ($gain > 0$) réinitialisent leur compteur à d ($counter_\alpha = 2$, $counter_\beta = 2$) tandis que les autres agents le décrémente de 1 ($counter_\gamma = 1$) (ligne 13). Par la suite, les agents échangent leur compteur avec leurs voisins afin de ne garder que la plus grande valeur du voisinage ($counter_\gamma = 2$).

Comme les compteurs ne sont pas nuls, une nouvelle itération commence. Les agents communiquent afin de mettre à jour leur modèle, ce qui permet d'obtenir les affectations et gains suivants :

$$\alpha \ : \ (o = 3) \qquad \beta \ : \ (p = 1, \, q = 2) \qquad \gamma \ : \ (r = 3) \tag{4.16}$$

$$g_\alpha = 3 \qquad \text{et} \qquad g_\beta = 0 \qquad \text{et} \qquad g_\gamma = 1 \tag{4.17}$$

À cette itération, l'agent β n'a pas changé d'affectation et possède donc un gain nul. Il ne gardera donc pas sa nouvelle affectation. À l'inverse, les agents α et γ ont les gains maximaux de leur voisinage, leur seul voisin β ayant un gain de 0. Ces deux agents garderont donc leur nouvelle affectation lors de cette itération et ainsi la nouvelle affectation jointe sera :

$$\alpha \ : \ (o = 3) \qquad \beta \ : \ (p = 1, \, q = 2) \qquad \gamma \ : \ (r = 3) \tag{4.18}$$

L'utilisation de cette affectation jointe permet d'atteindre l'état final suivant :

$$\{3, \, 0, \, 4, \, 0, \, 0\} \text{ avec comme valeur globale } C^2_\omega = 2. \tag{4.19}$$

La réduction des compteurs ou la réinitialisation des compteurs réalisée en fonction des gains conduit à $counter_\alpha = 2$, $counter_\beta = 1$ et $counter_\gamma = 2$. L'agent β a des voisins avec une valeur de compteur plus grande que la sienne, il gardera donc la valeur maximum ($counter_\beta = 2$). Aucun compteur n'a atteint 0, donc tous les agents commencent une nouvelle itération qui mènera aux affectations suivantes.

$$\alpha \ : \ (o = 3) \qquad \beta \ : \ (p = 1, \, q = 3) \qquad \gamma \ : \ (r = 3) \tag{4.20}$$

Ces affectations locales amènent aux gains suivant des agents :

$$g_\alpha = 0 \qquad \text{et} \qquad g_\beta = 1 \qquad \text{et} \qquad g_\gamma = 0 \tag{4.21}$$

Lors de cette $3^{\text{ème}}$ itération, seul l'agent β possède une amélioration non nulle. Il gardera donc sa nouvelle affectation tandis que les deux autres agents resteront avec leur affectation précédente.

La nouvelle affectation jointe est donc :

$$\alpha \ : \ (o = 3) \qquad \beta \ : \ (p = 1, q = 3) \qquad \gamma \ : \ (r = 3) \tag{4.22}$$

Cette nouvelle affectation jointe permet d'atteindre l'état suivant :

$$\{3, 1, 3, 0, 0\} \text{ avec comme valeur globale } C^3_\omega = 1. \tag{4.23}$$

Cet état correspond à une des configurations optimales pour un horizon de gestion de 1, mais les agents ne le détectent pas aussitôt, car leurs compteurs sont égaux à 2 ($counter_\alpha = counter_\beta = counter_\gamma = 2$).

Lors de l'itération suivante, aucun agent ne pourra améliorer son affectation locale puisqu'ils se trouvent tous dans un optimum local. Le gain de chaque agent sera donc nul : $g_\alpha = g_\beta = g_\gamma = 0$. De ce fait, tous les compteurs seront décrémentés de 1 ($counter_\alpha = counter_\beta = counter_\gamma = 1$) et aucun changement de politique n'est fait. L'itération suivante sera similaire et permettra à tous les agents de décrémenter leur compteur à 0 ce qui leur permet de détecter qu'ils ne peuvent plus s'améliorer, marquant ainsi la fin de la résolution. L'affectation calculée et l'état atteint en l'appliquant sont visibles sur la figure 4.6.

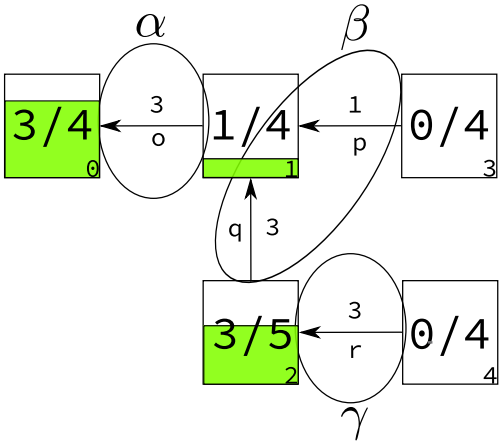

FIGURE 4.6 – Réseau d'entrepôts après l'application de l'affectation calculée (visible sur les arcs)

4.3 Comparaison de l'algorithme OCLP avec une approche optimale et une approche gloutonne

Des expérimentations ont été réalisées sur une application simple de la classe de problèmes considérée afin de comparer l'algorithme distribué proposé, OCLP, à un outil de modélisation couplé à une méthode d'optimisation centralisée de la littérature : les MDPs et *Value Iteration*. Une méthode gloutonne, distribuée sans coordination ni communication est aussi testée. Avec cette méthode, chaque agent construit sa politique locale en ignorant les autres agents. Elle est utilisée afin de mettre en avant l'impact de la coordination sur les solutions produites. L'algorithme *Value Iteration*, présenté en section 2.1.3 (page 28), est utilisé pour la résolution des MDPs locaux de chaque approche. Un facteur d'atténuation $\gamma = 1$ est utilisé, puisque la prise en compte du temps dans les états implique une borne sur le nombre d'itération pour calculer une politique optimale.

Les calculs ont été effectués sur un cluster, mis à notre disposition par l'IMT-Lille-Douai, composé de machines ayant chacune entre 24 et 32 Go de mémoire vive et de 8 cœurs (1 600 MHz). Les algorithmes ont été implémentés en Java, en utilisant le framework multi-agent JADE [Bellifemine et al., 1999] et une bibliothèque, développée en interne par l'équipe MAD au laboratoire GREYC à Caen, pour le traitement des MDPs.

Différents problèmes sont considérés (table 4.1), sur un horizon de prédiction infini ($\gamma = 0{,}9$). Ils sont définis par un ensemble de composants ayant une quantité de ressource définie dans un ensemble $\{0, \ldots, max\}$, où la quantité maximale max pouvant être stockée dépend du scénario. La capacité optimale d'un composant e, notée $goal_e$, est visible dans les nœuds sur les schémas des différents scénarios. Chaque composant est relié à d'autres composants par des effecteurs permettant le déplacement de la ressource. À chaque pas de temps, tous les arcs peuvent déplacer jusqu'à 3 unités de ressource du composant source vers le composant cible. L'effecteur dont la source n'est pas modélisée déplace un minimum d'une unité de ressource à chaque pas de temps. Par simplicité, les transferts de ressource sont considérés comme instantanés, simultanés et naturellement discrétisables. Ceci donne le potentiel aux ressources de traverser le système dans son ensemble en une seule itération.

Les déplacements de ressource ne sont pas déterministes. Lorsque de la ressource est déplacée, il existe un risque d'en perdre une certaine quantité. Plus la quantité déplacée est importante, plus les risques de pertes deviennent grands. Perdre de la ressource de cette façon n'induit pas de pénalité à l'agent responsable du transfert. Lorsqu'un composant est en surcapacité ou en sous-capacité, une grande pénalité (5 000) sera payée par les agents. Autrement, pour chaque agent e ne respectant pas sa capacité optimale, les agents paieront un coût quadratique :

$$(goal_e - capacitéActuelle_e)^2 \tag{4.24}$$

Les états locaux d'un composant correspondant à la surcapacité et à la sous-capacité sont notés respectivement $max + 1$ et -1. Lorsqu'ils sont atteints, après que la pénalité soit prise en

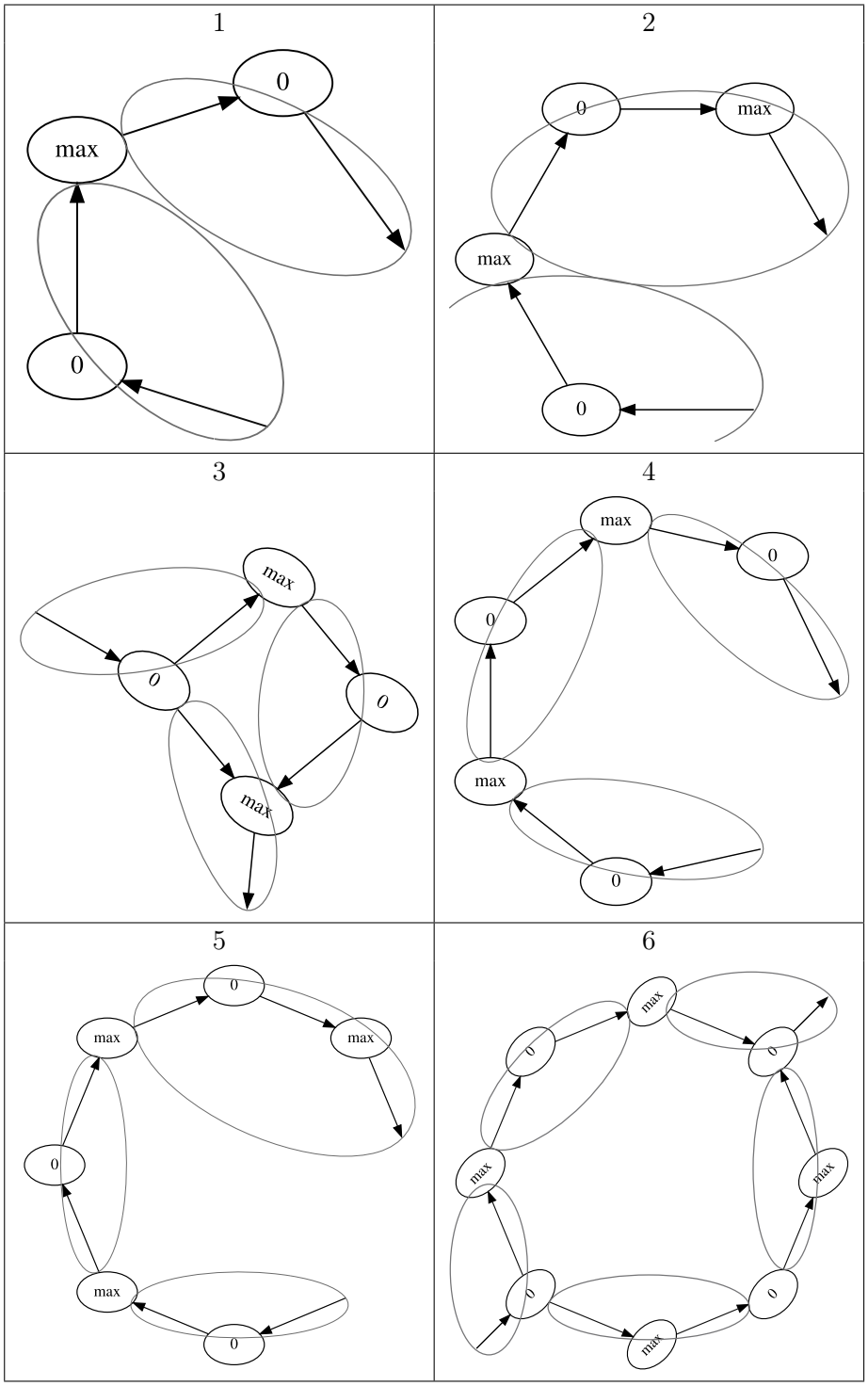

TABLE 4.1 – Représentation visuelle des scénarios et du partitionnement : les composants (ovales) sont reliés par des effecteurs (arcs). Les décompositions en agents sont définies par les groupes d'arcs.

compte, le composant retournera immédiatement dans l'état valide le plus proche (0 ou max). De ce fait, du point de vue de l'évolution du modèle, l'état local -1 (resp. $max+1$) est équivalent à l'état local 0 (resp. max). L'ensemble des états locaux d'un composant est $\{-1,0,\ldots,max,max+1\}$.

L'algorithme OCLP nécessite une décomposition en agents et une politique initiale pour chaque agent afin de commencer la résolution distribuée. Pour ces expériences, les politiques initiales des agents ont été définies arbitrairement par :

$$\pi_\alpha^0(s) = a_0 \qquad \forall s \in S_\alpha \text{ et } \forall \alpha \in Ag \tag{4.25}$$

où $a_0 \in A_\alpha$ correspond à l'action de l'agent α déplaçant le moins de ressource possible. Les décompositions en agents sont visibles sur la table 4.1. Ce partitionnement a été obtenu grâce à un algorithme heuristique présenté à la fin de ce chapitre. Ces décompositions en agents sont aussi utilisées pour l'approche gloutonne.

Tous les scénarios sont optimisés à l'aide des MDPs, de l'algorithme distribué OCLP et de l'approche gloutonne. Les tables 4.2 et 4.3 listent pour chaque scénario les principaux résultats obtenus : le temps nécessaire pour obtenir la politique finale et une évaluation de cette politique. L'évaluation d'une politique consiste à mesurer l'écart moyen des composants à leur valeur optimale à chaque pas de temps. Ainsi, une évaluation de 1 indique qu'en moyenne, à un pas de temps donné, un composant est à 1 unité de ressource de sa valeur optimale. Les actions étant stochastiques, les résultats sont une moyenne de 10 000 simulations d'une durée de 25 pas de temps. Si l'évaluation est supérieure à la valeur maximale du scénario alors, cela signifie que des pénalités ont été payées. Les scénarios ont été optimisés sous deux conditions différentes, avec une capacité maximale des composants de 3 (table 4.2) puis de 8 (table 4.3). La valeur maximale de 3 a été choisie pour permettre l'optimisation de scénario avec les MDPs. Tandis qu'une valeur de 8 a été choisie pour complexifier les modèles et ainsi permettre une meilleure comparaison des temps de calcul entre les approches OCLP et gloutonne. L'algorithme OCLP a convergé vers une solution localement optimale dans chacun de ces scénarios.

Comme attendu, il est possible de voir sur les tables 4.2 et 4.3 que les MDPs fournissent les politiques avec les meilleures évaluations, l'algorithme *Value Iteration* garantissant l'optimalité des politiques calculées. Cependant, cette approche est rapidement incapable de représenter les modèles considérés à cause des limitations de mémoire. De ce fait, moins de la moitié des problèmes ont pu être traités en utilisant des MDPs.

À l'inverse, les approches distribuées (OCLP et gloutonne) ont pu traiter l'ensemble des scénarios avec les décompositions spécifiées précédemment, sans souffrir de problèmes mémoires. Les résultats de l'algorithme OCLP ne sont pas optimaux, mais s'en approchent pour les scénarios 1 et 2 (table 4.2). Dans le cas du scénario 3 (table 4.2) et du scénario 1 (table 4.3), l'écart à la solution optimale est en moyenne d'une unité de ressource. Pour les autres scénarios, il est impossible de comparer les résultats de l'algorithme OCLP à une solution optimale, car elles

	Centralisé	Distribué	
	MDP	OCLP	Gloutonne
Scénario 1	temps : 28s éval : 1,211	7s 1,340	6s 5,207
Scénario 2	8min 57s 1,107	18s 1,189	12s 4,479
Scénario 3	3min 29s 0,588	10s 1,219	7s 5,681
Scénario 4	Out of Memory	14s 1,150	7s 30,910
Scénario 5	OoM	22s 1,080	11s 28,059
Scénario 6	OoM	10s 1,274	6s 9,253

TABLE 4.2 – Comparaison des différentes approches sur des problèmes simples ; temps : temps de calcul, éval : écart moyen à la valeur optimale. La capacité maximale est fixée à 3.

ne peuvent être obtenues au vu des conditions d'expérimentations définies. Néanmoins, tous les résultats de l'approche proposée, dans la table 4.2 (resp. la table 4.3), sont du même ordre de grandeur. Il est donc possible de les considérer comme satisfaisants. De tels écarts entre la solution optimale et de l'algorithme OCLP sont attendus, car les agents ne prennent en compte qu'une estimation des choix de leurs voisins et l'impact des autres agents n'est pris en compte que par transitivité. L'approche gloutonne, sans coordination ni communication, conduit globalement à des résultats de qualités bien plus faibles. Pour l'ensemble des scénarios de la table 4.2 un grand nombre de pénalités a dû être payé. Dans le cas où la capacité maximale est de 8 (table 4.3), quelques politiques jointes calculées ne nécessitent pas le paiement de pénalité. Cependant, elles restent loin d'une solution optimale, en doublant les écarts moyens de l'algorithme OCLP.

En comparant les temps de calcul des différentes méthodes, il est facile de remarquer que l'approche centralisée est de loin la plus lente. Sur des petits problèmes, les temps de calcul des deux approches distribuées peuvent être comparables, mais lorsque la taille des problèmes augmente la différence devient visible. Cette différence est due aux constructions et optimisations successives de MDPs par l'algorithme OCLP, alors que l'approche gloutonne ne construit et n'optimise qu'un seul MDP par agent. Il faut noter que chaque résolution distribuée du problème nécessite quelques secondes en début de calcul pour initialiser les communications entre agents.

Un dernier jeu de simulations a été réalisé afin de mettre en avant différents impacts possibles de la décomposition d'un problème en agents, quant à la qualité de la solution et du temps de nécessaire à son obtention. Pour cela, certains scénarios ont aussi été optimisés à l'aide d'une

	Centralisé	Distribué	
	MDP	OCLP	Gloutonne
Scénario 1	temps : 2min 04s éval : 1,607	10s 2,770	8s 5,704
Scénario 2	OoM	2min 16s 2,565	40s 4,927
Scénario 3	OoM	2min 11s 2,761	7s 5,660
Scénario 4	OoM	1min 36s 2,455	16s 29,251
Scénario 5	OoM	2min 53s 2,385	40s 26,028
Scénario 6	OoM	2min 10s 2,913	9s 3,799

TABLE 4.3 – Comparaison des différentes approches sur des problèmes plus imposants ; temps : temps de calcul, éval : écart moyen à la valeur optimale. La capacité maximale est fixée à 8.

décomposition complète, c'est-à-dire où chaque agent est défini par un seul arc. Les résultats obtenus avec un partitionnement maximal sont comparés (table 4.4) à ceux obtenus pour un partitionnement plus intuitif visible sur la table 4.1.

	OCLP	
	partitionnement maximal	partitionnement plus intuitif
Scénario 1 $max = 3$	temps : 7s éval : 1,399	7s 1,340
Scénario 2 $max = 8$	10s 2,640	2min 16s 2,565
Scénario 6 $max = 8$	11s 21,874	2min 53 2,385

TABLE 4.4 – Comparaison de deux partitionnements en agents avec l'algorithme OCLP

La table 4.4 met en avant trois effets possibles de la décomposition maximale par rapport à une décomposition plus intuitive. Dans le cas du scénario 1, avec une capacité maximale à 3, les deux partitions produisent des résultats semblables à la fois en temps de calcul qu'en évaluation de la politique jointe. Pour le scénario 2, avec une capacité maximale de 8, les évaluations sont de nouveau similaires, mais l'utilisation d'un partitionnement maximal permet de réduire significativement le temps de calcul. Finalement, pour le scénario 6, avec une capacité maximale de 8, l'algorithme proposé n'est pas capable de produire une solution évitant les pénalités

comparativement à une décomposition en agents plus intuitive.

Un partitionnement maximal permet de réduire la taille des modèles de chaque agent et ainsi le temps de calcul des politiques locales. L'augmentation du besoin de coordination semble moins important que le gain de calcul de chaque agent. Avec l'algorithme OCLP, la coordination se fait de proche en proche. Ainsi, plus les agents seront loin les uns des autres moins la coordination globale sera efficace. Or, dans les réseaux considérés dans la table 4.1, les capacités de régulations de ressource sont limitées. Par exemple, il n'existe qu'une seule arrivée de ressource dans le système et déplacer plus d'une unité de ressource à la fois est une tâche incertaine. De plus, les objectifs sont fortement contraints, les valeurs optimales étant aux bornes de l'intervalle de fonctionnement, l'influence de la coordination globale devient donc très importante.

Ainsi, l'approche OCLP fournit des résultats sous-optimaux, mais pouvant être considérés comme valides pour la gestion de systèmes de grandes dimensions dont l'obtention d'une solution optimale peut s'avérer trop coûteuse. Elle permet un calcul plus rapide et moins coûteux en mémoire que les approches centralisées, car elle bénéficie pleinement des caractéristiques de la classe de problèmes traitée grâce à une résolution distribuée et coordonnée. Néanmoins, la qualité de la solution dépend de la décomposition en agent. Cette décomposition reste une tâche non triviale. Un algorithme de décomposition, utilisé pour calculer le partitionnement dit plus intuitif des problèmes de cette section, est proposé dans la section suivante.

4.4 Algorithme heuristique de décomposition en agent

Un algorithme heuristique a été proposé pour choisir les décompositions en agents du problème. Cet algorithme vise à trouver le meilleur partitionnement en k agents possible en un temps limité. L'obtention d'un partitionnement optimal selon une liste de critères prédéfinis est une tâche difficile et coûteuse en temps. De ce fait, un algorithme de partitionnement non optimal a été proposé (algorithme 5), il est inspiré d'une approche pour le partitionnement de cartes [Lozenguez et al., 2012].

L'algorithme 5 nécessite un k-partitionnement initial qui peut être quelconque. Ce partitionnement subira des perturbations successives dans le but d'améliorer la qualité des partitions. Dans ce cadre, une perturbation consiste à retirer aléatoirement un certain nombre d'effecteurs des partitions et de les replacer de façon optimale dans les partitions existantes, selon une heuristique. Ces perturbations seront appliquées jusqu'à ce qu'une des conditions d'arrêt soit atteinte : soit liée à une durée (en secondes ou en itérations), soit liée à une absence d'amélioration de qualité du partitionnement pendant une durée fixée.

Le nombre d'effecteurs déplacés à chaque itération est choisi aléatoirement dans l'intervalle $[1,\lceil \frac{m}{k} \rceil]$. La borne maximale $\frac{m}{k}$ est le ratio entre le nombre d'effecteurs m et le nombre de partitions k. Elle est utilisée pour limiter le nombre de changements possibles en une itération. Ainsi, l'obtention du meilleur partitionnement, après une perturbation (lignes 6-7), reste rapide.

Varier le nombre d'effecteurs déplacés par une perturbation permet de réaliser des transformations complexes des partitions. Par exemple, trois effecteurs sont répartis dans trois partitions différentes. La solution optimale serait d'avoir les trois dans une même partition. Cependant, n'en avoir que deux au même endroit est pénalisé. En les déplaçant un par un, il ne serait pas possible de regrouper les effecteurs dans une même partition. Il faudrait les regrouper un à un, or cela impliquerait une étape intermédiaire où deux effecteurs sont dans une même partition. Cette configuration réduisant fortement la qualité de la solution elle ne serait jamais atteinte et empêcherait le partitionnement idéal d'être choisi. Cependant, si deux effecteurs peuvent être déplacés en une seule fois, atteindre la solution optimale reste possible.

Plusieurs méthodes sont possibles pour choisir le nombre d'effecteurs à déplacer (ligne 3) à chaque itération. Une première consiste simplement à choisir aléatoirement à chaque itération un nombre j d'effecteurs à déplacer dans l'intervalle spécifié. Une seconde consiste à changer le nombre j à intervalle de temps/itérations fixés en suivant un ordre particulier (par exemple croissant/décroissant).

Algorithme 5 Partitionnement automatique en agents

1: Création d'un k-partitionnement initial
2: **répéter**
3: Choisir $j \in [1, \lceil \frac{m}{k} \rceil]$
4: Choisir aléatoirement j actionneurs
5: Retirer les actionneurs choisis de leurs partitions
6: Lister les assignations possibles des actionneurs dans les partitions
7: Garder le meilleur partitionnement produit selon l'heuristique
8: **jusqu'à** qu'une condition d'arrêt soit atteinte

L'intérêt d'un partitionnement par rapport à un autre est mesuré par des fonctions heuristiques. Ces fonctions ont pour but d'évaluer les partitions individuelles ou les combinaisons de partitions. Dans tous les cas, les heuristiques pénalisent très fortement les partitions vides. Un exemple d'heuristique, qui est celui utilisé pour l'obtention de tous les partitionnements de cette thèse, consiste à minimiser le nombre de valeurs à stocker dans les fonctions de transitions des agents. Cette heuristique vise à minimiser la taille des fonctions de transition et donc à minimiser le nombre de composants affecté par chaque agent. Ceci permet de limiter le nombre d'agents ayant connaissance d'un composant et ainsi de maximiser la probabilité que chaque agent contrôle un réseau connexe tout en minimisant l'espace mémoire requis.

Une discussion sur l'impact de la décomposition sur le temps de calcul et sur la qualité des politiques sera réalisée dans le chapitre suivant.

4.5 Conclusion

L'optimisation de la gestion d'un système distribué est une tâche complexe qui, à notre connaissance, se base le plus souvent sur une exploitation des caractéristiques spécifiques des modèles pris en compte. Ainsi l'algorithme OCLP proposé dans ce chapitre exploite la distribution naturelle des systèmes en composants, avec une interdépendance limitée des composants les uns aux autres. Cet algorithme assure l'arrêt lorsque tous les agents sont dans des optimums locaux, mais n'offre pas de garantie quant à la capacité de converger vers ces solutions et donc sur la capacité à s'arrêter.

L'approche proposée étant multi-agents, elle se base donc sur une distribution du système, posant ainsi un problème de décomposition. Une méthode heuristique a été définie afin d'obtenir des décompositions selon un ensemble de critères. La décomposition en agents aura une influence sur la qualité des résultats pouvant être obtenus ainsi que sur la durée nécessaire à leur obtention. Cet impact dépendra des problèmes traités.

Une comparaison de la méthode proposée avec une approche centralisée utilisant les MDPs ainsi qu'avec une méthode distribuée sans coordination des agents a été effectuée sur des scénarios illustratifs. La méthode centralisée testée fournit des résultats optimaux, mais souffre de temps de calcul élevés et requiert un espace mémoire important. Le passage à l'échelle d'une telle approche est donc impossible. L'algorithme OCLP permet d'obtenir des solutions sous-optimales, mais qui permettent de maintenir des écarts limités envers les objectifs de gestion, pour des problèmes ne pouvant être résolus de façon optimale dans des conditions de calcul équivalentes par des MDPs.

Dans le chapitre suivant, des résultats d'utilisation de l'algorithme OCLP sont présentés dans le cadre de la gestion de la ressource en eau dans les voies navigables.

Chapitre 5

Gestion sous incertitudes de la ressource pour les voies navigables

La gestion de la ressource en eau sur un réseau de voies navigables a pour but de conserver les conditions de navigation sur l'ensemble du réseau. Chaque bief a pour objectif de maintenir son rectangle de navigation et de minimiser l'écart à son niveau idéal : le NNL. Pour cela, le niveau d'eau des biefs considérés doit être optimisé au cours du temps. L'approche OCLP est donc utilisée pour atteindre cet objectif. Plusieurs expérimentations ont été réalisées afin de mesurer son efficacité sur la gestion de la ressource en eau de réseaux de voies navigables sous incertitudes. Ce chapitre reprend en partie des résultats précédemment publiés [Desquesnes et al., 2017c] [Desquesnes et al., 2018b].

Dans chaque cas, les politiques initiales sont construites de la même façon, en considérant que chaque agent transfère toujours le plus faible volume possible sur chaque point de transfert qu'il contrôle. Cette politique a l'avantage de correspondre, le plus souvent, à une mauvaise gestion de l'ensemble des biefs, c'est-à-dire à un non respect des rectangles de navigation des biefs. De cette façon, les agents sont obligés d'améliorer leur politique locale et de se coordonner pour obtenir des résultats satisfaisants.

Dans un premier temps, un réseau factice de 7 biefs similaires est considéré. Par la suite, un réseau réel de 3 biefs des voies navigables du réseau des Hauts-de-France, ainsi que son extension à 7 biefs sont traités selon divers scénarios d'utilisation présents et futurs. À cause d'un manque d'informations, certains débits et estimations de la demande de navigation de l'extension à 7 biefs ont été approximés à partir des données existantes. Ces scénarios sont, en partie, définis à partir des prévisions d'augmentation du trafic et du changement climatique [Chauveau et al., 2013]. Les politiques obtenues sont testées sur des simulateurs.

Les expérimentations ont été réalisées sur un cluster de machines, mis à notre disposition par l'IMT-Lille-Douai. Chaque machine possède 48 Go de mémoire avec 12 cœurs (1600 MHz) pour la gestion du réseau factice, et 64 Go de mémoire de 32 cœurs ($>$ 1200 MHz) pour celles

sur des réseaux des Hauts-de-France. Chaque agent a le contrôle exclusif d'une machine pour l'optimisation de sa politique locale. La construction des fonctions de transition est parallélisée de façon à utiliser au maximum les capacités des machines. Les discrétisations en intervalles des états et des actions ont été choisies de façon à ce que la taille de la fonction de transition représentée en Java ne nécessite pas plus de la moitié de l'espace mémoire disponible sur les machines. Les décompositions en agents suivent la même logique.

5.1 Application sur un réseau factice de voies navigables

5.1.1 Présentation du réseau

Un réseau factice de 7 biefs et de 14 points de transferts (vannes, écluses, barrages, ...) a été créé (figure 5.1). Les biefs sont représentés par des carrés avec leurs objectifs et leurs intervalles autorisés de gestion spécifiés à l'intérieur de chaque carré. Les arcs représentent les points de transferts orientés dont la capacité (volume minimal et maximal transférable pour chaque pas de temps) est spécifiée par un intervalle. Chaque bief est divisé en 8 intervalles de volumes : 6 de taille finie et les 2 extremums considérés de taille infinie, comme spécifié dans la section 3.4 (page 61), tandis que les points de transferts (effecteurs) sont divisés en intervalle de taille 5. Ce choix de discrétisation a été effectué dans le but de minimiser le nombre d'états et d'actions tout en minimisant les variations liées à la taille des intervalles. La planification se fait sur une durée de 8 pas de temps de 12 heures. L'utilisation de fausses bassinées (i.e. utiliser une écluse dans le seul but de déplacer de l'eau) est actuellement une technique que les gestionnaires évitent, notamment durant la nuit. De ce fait, les modélisations de réseaux suivantes ne confèrent pas aux écluses la possibilité de déplacer de l'eau hormis pour permettre la navigation.

Dans le cadre d'une modélisation mono-agent avec MDP : le nombre d'états est de $|S| = 8^7 \times (8+1) \approx 10^7$, tandis que le nombre d'actions est de $|A| = 3^6 \times 4^2 \times 6^6 \approx 10^9$. Ainsi la fonction de transition d'un tel modèle représentée par une matrice creuse (i.e. une matrice ne stockant pas les valeurs nulles) nécessiterait de stocker $|S| \times |A| \times |S| \times \frac{1}{\tau+1} \approx 10^{22}$ valeurs. La fraction $\frac{1}{\tau+1}$ est présente, due à la linéarité du temps. Un état à un instant $t \in [0,\tau]$ ne peut atteindre qu'un seul état au pas de temps suivant $t+1$. Cependant, à cause des limitations d'espace mémoire et de temps de calcul, il ne nous a pas été possible de tester une décomposition avec un seul agent qui aurait permis une comparaison avec une solution optimale en considérant le problème de manière centralisée.

5.1.2 Présentation de la méthodologie

L'algorithme introduit est ainsi testé sur ce réseau. Pour cela, plusieurs décompositions du réseau en agents ont été proposées grâce à l'algorithme décrit dans la section 4.4 (page 86). Les décompositions en agent vont de 14 agents, le maximum, à 6 agents, le plus petit nombre d'agents pouvant résoudre le problème avec la configuration utilisée. Dans ces expérimentations,

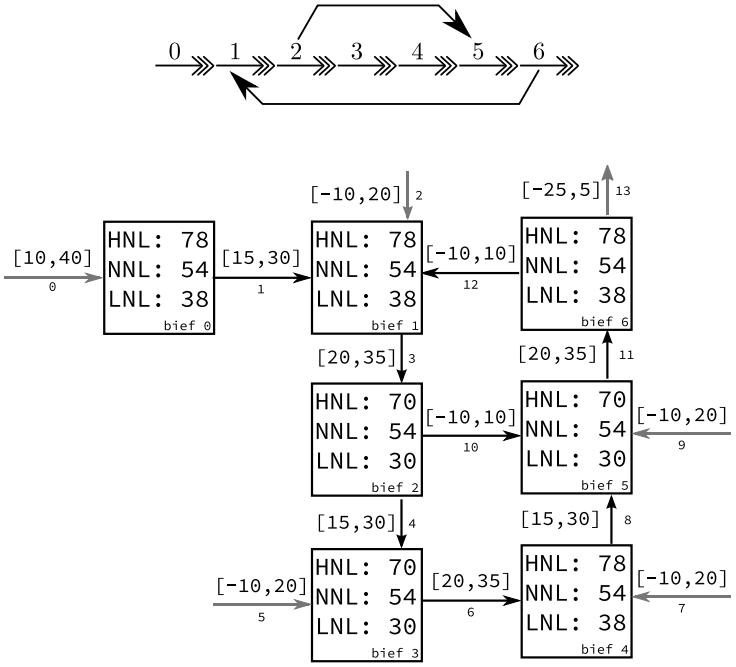

FIGURE 5.1 – Scénario fictif de 7 biefs (carré) et 14 points de transfert (arcs)

la discrétisation en états et en actions est identique pour chaque décomposition, ces dernières ont pour but de permettre le meilleur passage à l'échelle de la discrétisation choisie. Pour comparer les politiques jointes obtenues pour chaque décomposition, plusieurs critères sont observés. Premièrement, le temps de calcul nécessaire pour l'obtention de la politique jointe ainsi que l'espace mémoire de la fonction de transition jointe, c'est-à-dire le nombre d'éléments stockés par les matrices de transition de chaque agent. Les autres indicateurs correspondent à la qualité de la politique jointe obtenue. Ils indiquent le pourcentage moyen du temps où un bief est hors de son rectangle de navigation (out) et le pourcentage moyen d'écart des biefs par rapport à leur intervalle de navigation (avg). L'écart moyen d'un bief à ses bornes de son rectangle de navigation, à un instant donné, est défini par une échelle proportionnelle où 0% correspond au NNL et où 100% correspond soit au HNL soit au LNL. Les deux derniers critères (out et avg) sont obtenus lors des simulations du réseau utilisant les politiques jointes. Ces simulations sont stochastiques, car les actions étant des intervalles de volume, le volume à transférer par chaque effecteur est choisi dans cet intervalle de définition. Afin d'avoir une vision sur l'ensemble de l'intervalle, le volume choisi est tiré aléatoirement dans l'intervalle choisi par la politique. Pour cela, les deux derniers critères sont des moyennes sur 50 000 simulations utilisant la politique jointe et en commençant dans l'état idéal (tous les niveaux sont au NNL). L'état initial d'une simulation est toujours le même, de façon à ce que les résultats de deux simulations soient comparables.

décomposition	nombre de transitions possibles	temps (s)	out (%)	avg (%)
6 agents	$5{,}799\,936 \times 10^6$	591	0,000	17,13
7 agents	$2{,}850\,816 \times 10^6$	167	0,001	15,38
8 agents	$4{,}751\,36 \times 10^5$	68	0,000	16,45
9 agents	$4{,}345\,60 \times 10^5$	35	0,000	17,27
10 agents	$3{,}530\,24 \times 10^5$	32	0,021	17,16
11 agents	$2{,}919\,68 \times 10^5$	30	0,023	17,83
12 agents	$2{,}309\,12 \times 10^5$	31	0,016	17,27
13 agents	$1{,}698\,56 \times 10^5$	30	0,009	16,64
14 agents	$1{,}088\,00 \times 10^5$	26	0,007	15,57
gloutonne 7	$2{,}850\,816 \times 10^6$	4	6,480	37,34

TABLE 5.1 – 7 biefs avec décomposition variable

5.1.3 Analyse des résultats obtenus

Les résultats présentés sur la table 5.1 mettent en avant une réduction attendue du temps de calcul et de l'espace mémoire nécessaire pour trouver une politique jointe lorsque le nombre d'agents augmente. À partir d'un certain point, ces gains deviennent négligeables. Avec toutes les décompositions, il est très rare qu'un bief se trouve hors de son rectangle de navigation. De même, l'écart moyen d'un bief à son NNL à chaque pas de temps se situe entre 15% et 18%. Cela peut être considéré comme acceptable par rapport à la discrétisation choisie pour ce réseau. Cela signifie en effet que les biefs se trouvent en moyenne dans l'intervalle contenant le NNL.

Contrairement à ce qui pourrait être attendu, augmenter le nombre d'agents ne réduit pas forcément l'écart moyen au NNL en utilisant la politique jointe obtenue. Néanmoins, les décompositions de moins de 10 agents n'ont pas, ou très rarement, des biefs sortants de leur rectangle de navigation contrairement aux décompositions de tailles supérieures, à l'inverse des décompositions plus importantes. Lorsque le nombre d'agents augmente, la portion du réseau visible par chaque agent diminue. Chaque agent a ainsi une plus grande dépendance envers les approximations des plans de ses voisins. Cela devrait diminuer la qualité de la politique globale, lorsqu'une coordination de l'ensemble des agents est nécessaire. Cependant, moins un agent contrôle d'effecteurs moins il observera de composants et ainsi il sera fait abstraction de moins d'information lors de l'adaptation des politiques. Cet agent pourra alors informer plus précisément ses voisins sur ses actions par rapport à leurs connaissances partagées. Il s'agit donc d'un compromis entre la capacité à optimiser localement le réseau et l'optimiser dans sa globalité.

À titre de comparaison, la meilleure politique obtenue par une résolution gloutonne (pour 7 agents) a un écart moyen de 37,34% au NNL et les biefs sont hors de leur rectangle de navigation 6,48% du temps, soit environ 3 000 fois plus qu'avec l'algorithme OCLP.

Il est possible de voir sur la figure 5.2, l'évolution du volume de chaque bief en suivant la

politique produite pour la décomposition en 7 agents. L'échelle des volumes est normalisée de façon à ce qu'une valeur 0 corresponde au NNL de chaque bief et à ce que les valeurs -100 et 100 correspondent respectivement aux LNL et aux HNL des biefs. Cette simulation utilise la valeur moyenne de chaque intervalle d'action choisi par la politique comme volume à transférer. Similairement à la table 5.1, l'écart au NNL est faible avec un écart maximum de 30% et un écart moyen sur la durée de 10%. Un effet d'oscillation, sans divergence, des niveaux autour des niveaux optimaux peut être observé. L'amplitude de ces oscillations va de pair avec la finesse de la discrétisation. En tirant des valeurs aléatoires dans les intervalles des actions choisies par la politique, l'écart type sur les courbes serait d'environ 13%.

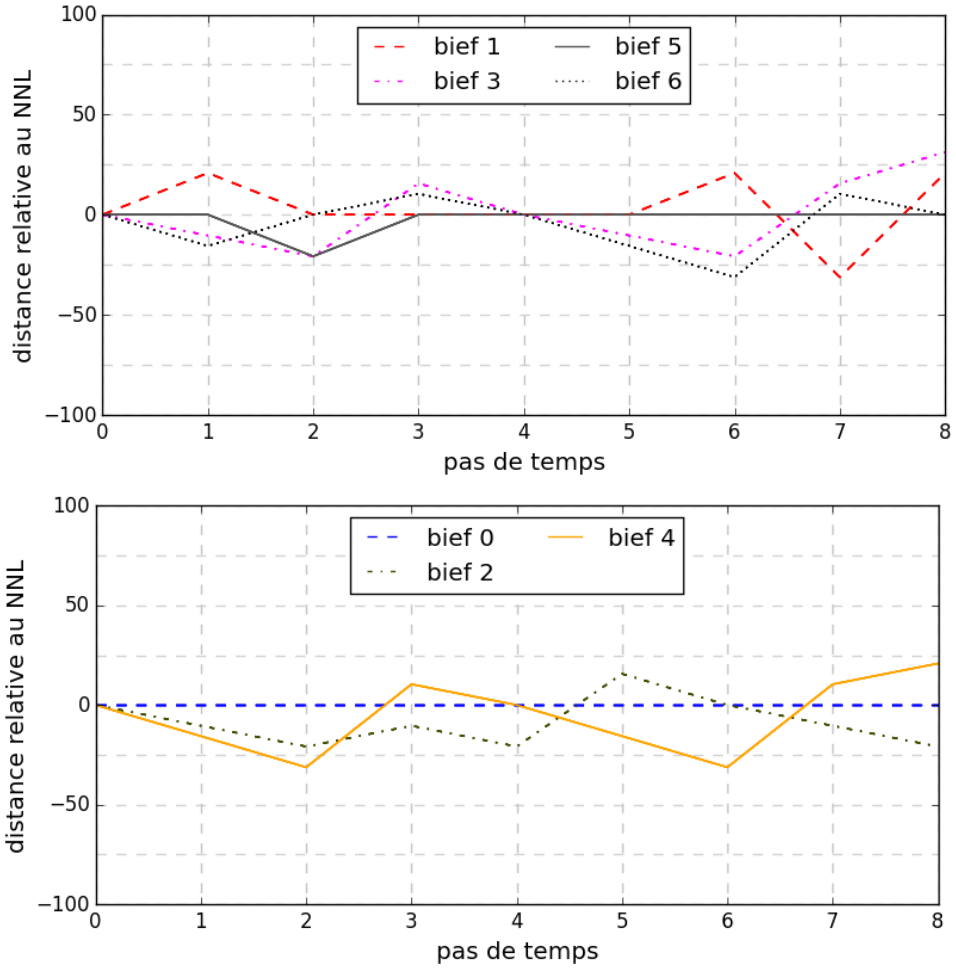

FIGURE 5.2 – Évolution de la distance relative au NNL sur la durée avec la décomposition en 7 agents, avec des pas de temps de 12 heures

Les résultats obtenus sur ce scénario et ses décompositions sont bons et permettent aux biefs de rester relativement proches de leur niveau objectif sans sortir de leur rectangle de navigation.

Cependant, l'application de la politique reste délicate. Il est important et intéressant de noter qu'il a été remarqué expérimentalement qu'un usage optimal des intervalles choisis par la politique permettrait de maintenir le NNL de tous les biefs à chaque instant, dans le cas de la décomposition en 7 agents. Cependant, déterminer la valeur optimale à choisir dans un intervalle est un problème difficile et a un impact non négligeable sur la qualité de la gestion. Il s'agit en effet d'une version simplifié du problème résolu, avec des domaines d'actions réduits. Des solutions potentielles ont été envisagées et sont présentées à la fin de ce chapitre et en perspective de cette thèse.

5.2 Application sur un réseau réel de 3 biefs

FIGURE 5.3 – Position du réseau Douai–Fontinettes–Grand-Carré sur une carte du Nord-Pas-de-Calais

Le sous-réseau Douai–Fontinettes–Grand-Carré (figure 5.3) est une sous-partie du réseau de voies navigables du nord de la France qui a été modélisée suivant l'approche précédemment décrite. Ce réseau est à la croisée de plusieurs bassins versants importants des Hauts-de-France. Il est composé de trois biefs couvrant une distance d'environ 90 kilomètres. Les informations concernant la bathymétrie et les débits de ce réseau correspondent à la réalité et ont été obtenues grâce à l'aide des gestionnaires des Voies Navigables de France. Les objectifs de gestion et leur équivalent en volumes sont fournis dans la table 5.2. Le bief Cuinchy-Fontinettes (bief 1) est tout particulièrement important. Puisque, de part sa configuration, sa gestion est particulièrement

contrainte. Les trois biefs, représentés par des cercles, sont interconnectés par les douze points de transferts (vannes, écluses, barrages, ...), représentés sur la figure 5.4 par les arcs. Par exemple, le point de transfert 7, lié au bief 1, correspond à l'écluse des Fontinettes. Ces biefs sont aussi connectés à une partie non modélisée du réseau (points de transfert 0, 1, 2, 7, 8, 9, 10 et 11). Cette partie non modélisée peut correspondre à des rivières naturelles, mais aussi à des biefs gérés par un ou plusieurs agents externes. L'état des biefs est divisé en un nombre variable d'intervalles de volume de taille finie et, comme précédemment, en 2 intervalles de taille infinie. La taille des intervalles de chaque bief et leur nombre sont spécifiés dans la table 5.3. Cette discrétisation est obtenue en utilisant le raisonnement présenté dans la section 3.4.2 (page 63). Dans ce scénario, seuls trois points de transfert sont contrôlables par les gestionnaires (4, 6 et 11). Ils ont été discrétisés respectivement en 124, 375 et 352 actions, couvrant la plage de fonctionnement de ces ouvrages. Il s'agit de vannes contrôlables. Sur ce réseau, l'écart de niveau entre le NNL d'un bief et le niveau du LNL (resp. HNL) est de 5 centimètres, à l'exception du bief Douai-Don-Cuinchy où l'écart de niveau entre le NNL et le HNL est de 10 centimètres.

Bief	Nom	LNL (m^3)	NNL (m^3)	HNL (m^3)
0	Douai-Don-Cuinchy	8654381	8772934	9010040
1	Cuinchy-Fontinettes	9348300	9458280	9568260
2	Don–Grand-Carré	3766098	3824038	3881978

TABLE 5.2 – Propriétés du réseau Douai–Fontinettes–Grand-carré

Bief	Taille d'intervalle (m^3)	Nombre d'intervalles
0	39544	12
1	15800	17
2	11588	13

TABLE 5.3 – Discrétisation du réseau Douai-Fontinettes-Grand carré

Plusieurs scénarios d'opération correspondant à des situations réelles et prévues ont été proposés pour tester l'approche de planification conçue. Tous ces scénarios se déroulent sur une durée de 7 périodes de 12 heures. Cela correspond à une demi-semaine de gestion. Le trafic journalier passant par une écluse est une estimation du nombre de sassées (utilisation d'écluse) sur une période de navigation (table 5.4). Par simplicité de modélisation, le volume d'eau déplacé à chaque utilisation d'une écluse est considéré comme constant et son déplacement comme immédiat. Par rapport au pas de temps de 12 heures, cette hypothèse n'est pas restrictive puisque une sassée dure de 10 à 30 minutes. Les capacités minimales et maximales de chaque point de transfert contrôlables sont constantes sur l'horizon de prédiction et sont définies dans la table 5.5. Les volumes transférés par les autres points de transfert sont dépendants des scénarios et sont introduits dans la description des scénarios.

Les scénarios utilisés ici sont faiblement impactés par les incertitudes lors de la planification.

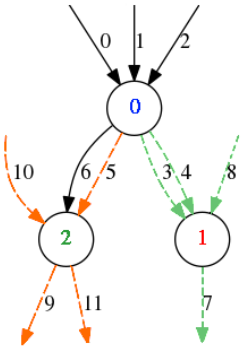

FIGURE 5.4 – Sous-réseau de 3 biefs et sa décomposition en 3 agents des 12 points de transfert

Seul un des quatre scénarios utilisés comporte des incertitudes liées à un possible événement de pluie affectant un des biefs. La décomposition du réseau en agents est visible sur la figure 5.4 grâce à un code couleur. Chaque agent contrôle des arcs de couleurs et de tracés différents.

Écluse	0	3	5	7	9
Trafic (nombre de sassées)	21	13	14	10	16
Volumes (m^3)	140 889	45 838	82 656	230 000	117 424

TABLE 5.4 – Trafic moyen sur une période de navigation de 12 heures et volumes déplacés par chaque utilisation de l'écluse

Point de transfert	4	6	11
Volumes (m^3)	[0, 432 000]	[0, 1 296 000]	[0, 2 592 000]
Taille d'intervalle (m^3)	3 484	3 456	7 364

TABLE 5.5 – Capacités minimales et maximales de points de transfert contrôlables sur une période de 12 heures

À chaque pas de temps la politique d'un agent détermine une action à effectuer : un intervalle de volume pour chaque point de transfert. Les simulations effectuées utilisent des valeurs tirées aléatoirement dans ces intervalles d'actions choisis par la politique plutôt que de prendre la valeur moyenne ou toute autre valeur spécifique. Puisque les volumes transférés sont choisis aléatoirement dans les intervalles de la politique, cinq simulations ont été faites pour chaque scénario. Cela aide à visualiser les variations des résultats liées à la discrétisation, tout en laissant les figures lisibles.

Pour résumer, quatre scénarios avec des conditions de gestion différentes ont été planifiés :

1. situation normale ;
2. situation d'étiage : les volumes non contrôlables sont réduits ;
3. situation future : augmentation du trafic et navigation autorisée la nuit[6] ;
4. situation de pluie : un événement de pluie intense est prévu (60% de chance d'occurrence) sur un des biefs.

Une seule politique est calculée pour chaque scénario. Elle est utilisée dans différentes simulations. Chaque scénario est simulé avec différentes conditions d'évolution du réseau telles que :

1. l'état initial de chaque bief est son NNL ;
2. l'état initial de chaque bief est proche de son HNL ;
3. l'état initial de chaque bief est proche de son LNL ;
4. le trafic est 10% plus important que prévu ;
5. le trafic est 10% moins important que prévu.

Cela donne finalement 20 résultats d'expérimentation exposés par la suite. Une dernière expérimentation sera effectuée en reprenant le premier scénario tout en prenant en compte les incertitudes connues du réseau. L'impact de la prise en compte des incertitudes sera discuté.

5.2.1 Scénario 1 : conditions normales

Le premier scénario correspond à des conditions normales de navigation. La navigation n'est autorisée que durant la journée (12 heures). Aucune perturbation n'est prévue sur la durée de planification. Les volumes échangés au niveau des points de transfert non contrôlés par les gestionnaires dans ce scénario sont spécifiés dans la table 5.6. Les déplacements de ces volumes non contrôlés sont considérés comme constants sur la durée de la simulation. La politique jointe a été obtenue en 20 minutes. Elle est unique pour toutes les situations considérées.

Point de transfert	1	2	8	10
Volumes (m^3)	283 392	−43 200	27 216	51 840
Débit (m^3/s)	6,56	−1,00	0,63	1,20

TABLE 5.6 – Volumes déplacés par les points de transfert incontrôlables sur une période de 12 heures

Sur la figure 5.5, l'évolution du volume de chacun des trois biefs est représentée relativement à leur rectangle de navigation sur l'horizon de prédiction. Les valeurs de 100, 0 et −100 correspondent respectivement aux HNL, NNL et LNL du bief. Le rectangle de navigation a été

6. Pour rappel, actuellement la navigation est uniquement autorisée le jour, ce qui permet de bénéficier de la nuit pour laisser les biefs se remettre à niveau.

normalisé afin d'améliorer la représentation des résultats. Pour chaque bief, le volume oscille autour du NNL avec un écart qui, à de rares exceptions près, est inférieur à 25%. Dans ce cas, un écart de 25% au NNL est équivalent à une variation de 1,25 centimètres sur les biefs 1 et 2 et entre 1,25 et 2 centimètres sur le bief 0. Les oscillations du niveau d'eau sont naturellement présentes à cause de l'alternance des périodes de navigation. Elles sont amplifiées par la discrétisation des états et des actions en intervalles.

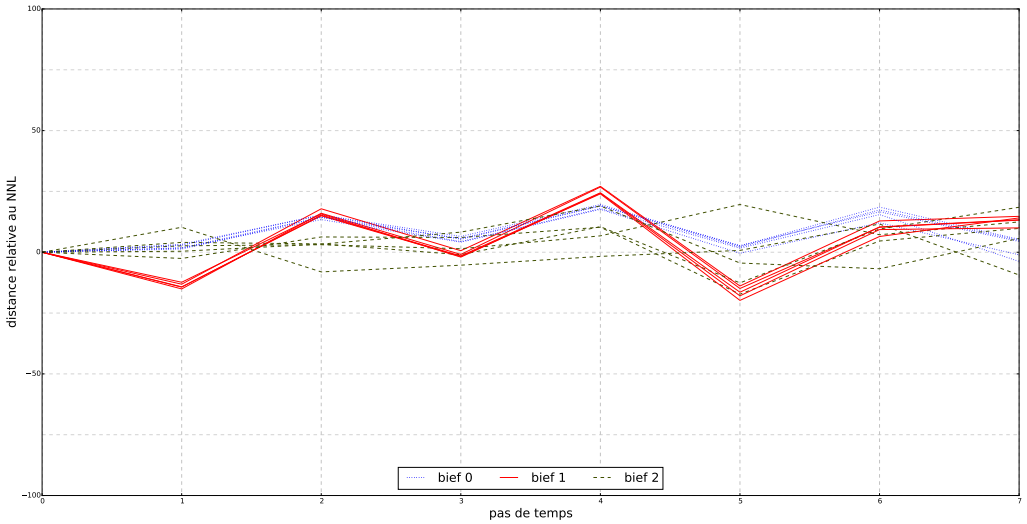

FIGURE 5.5 – Gestion du réseau sous des conditions normales avec les NNL comme états initiaux, évolution du volume des biefs relatif à leur rectangle de navigation avec un pas de temps de 12 heures

L'état d'un bief est représenté par un intervalle de volume. Or l'action optimale à effectuer sur ce bief peut dépendre de la valeur réelle dans cet intervalle. Par exemple, si un intervalle englobe le NNL alors l'action optimale pour la partie supérieure de l'intervalle serait de baisser le volume alors que l'inverse serait souhaitable pour la partie inférieure. Cet effet est d'autant plus important avec la discrétisation des actions. Ainsi une discrétisation plus fine en états et en actions devrait permettre de réduire l'intensité de ces oscillations, mais augmenterait de façon non négligeable la taille du modèle et donc le temps de calcul.

La figure 5.6 représente les volumes d'eau qui sont déplacés à chaque pas de temps en suivant la politique de gestion calculée. Elle correspond à une des consignes permises par les intervalles d'actions, ici en choisissant les valeurs moyennes. Les effecteurs 1, 2, 8 et 10 transfèrent la même quantité d'eau à chaque pas de temps et ne sont donc pas représentés ici. Pour chaque effecteur contrôlable (4, 6 et 11), la valeur maximale pouvant être transférée est spécifiée par une ligne rouge. Cependant, ici l'objectif reste de montrer l'évolution du volume de chaque bief par rapport à son rectangle de navigation et à son NNL et non les actions ou consignes permettant d'y arriver.

Dans le cas où les biefs commencent dans une position sous-optimale proche du HNL ou du

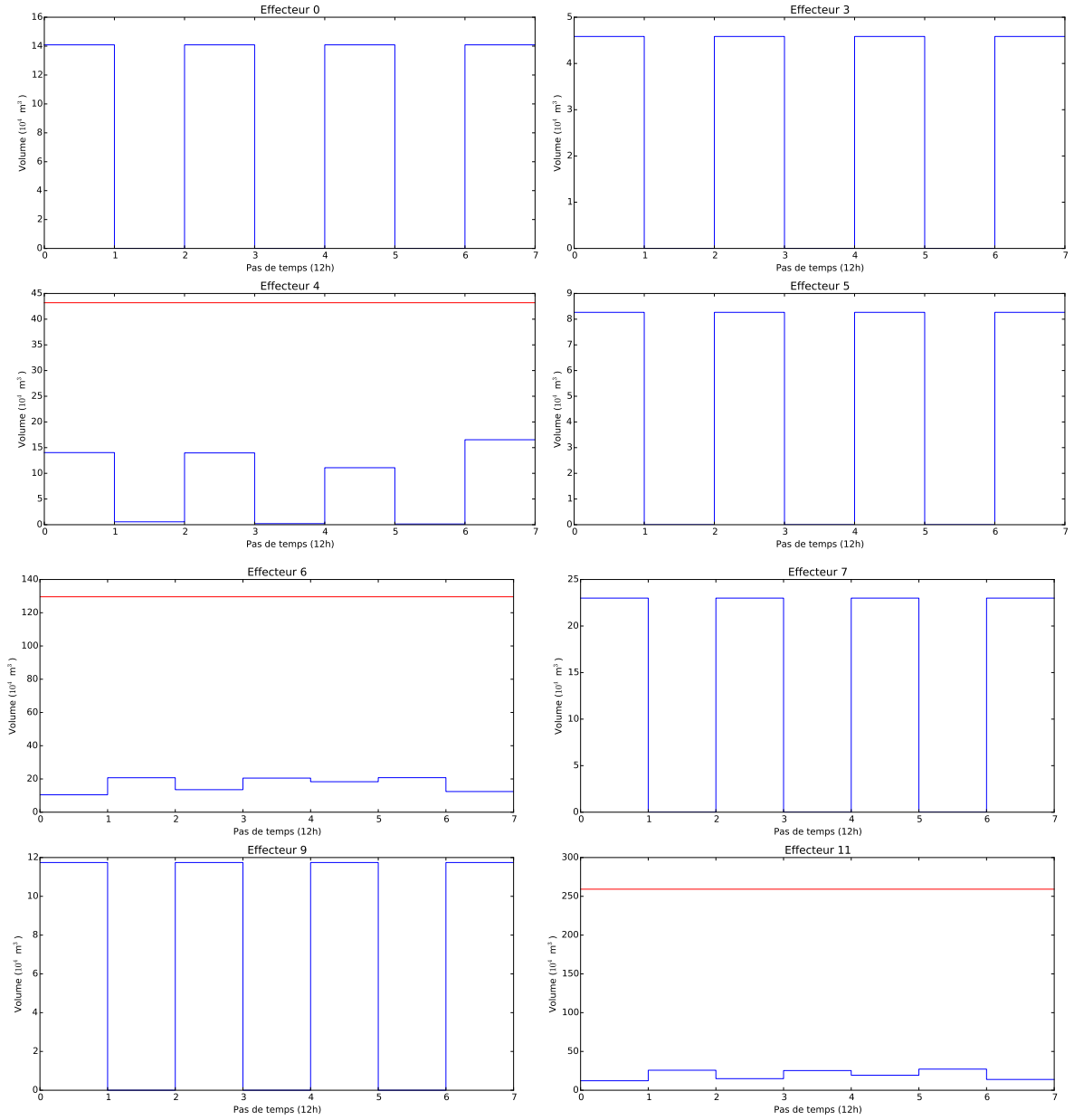

FIGURE 5.6 – Volumes (10^4 m^3) déplacés par les effecteurs durant chaque pas de temps (12 h), lors d'une simulation sous des conditions normales avec les NNL comme états initiaux. En rouge, les valeurs maximales transférables par les effecteurs contrôlables. Les effecteurs déplaçant le même volume à chaque instant ne sont pas représentés

LNL (figures 5.7 et 5.8), l'utilisation de la politique jointe permet de récupérer et d'atteindre une solution proche du NNL avant la fin de la simulation. Dans le cas d'un événement ayant

approché le réseau de ses LNL, voir la figure 5.8, la récupération est relativement rapide. Seules deux périodes sont nécessaires à la politique jointe obtenue pour remettre le réseau proche de son NNL. Dans le cas inverse illustré par la figure 5.7, les résultats sont similaires. Il peut être remarqué, sur ces trois premiers jeux de simulation, que l'évolution du bief 1 est toujours dégradée lors de la cinquième période, alors que l'ensemble du réseau est dans une configuration valide. Néanmoins, cet écart est pris en compte dans la planification et est compensé sur les périodes suivantes.

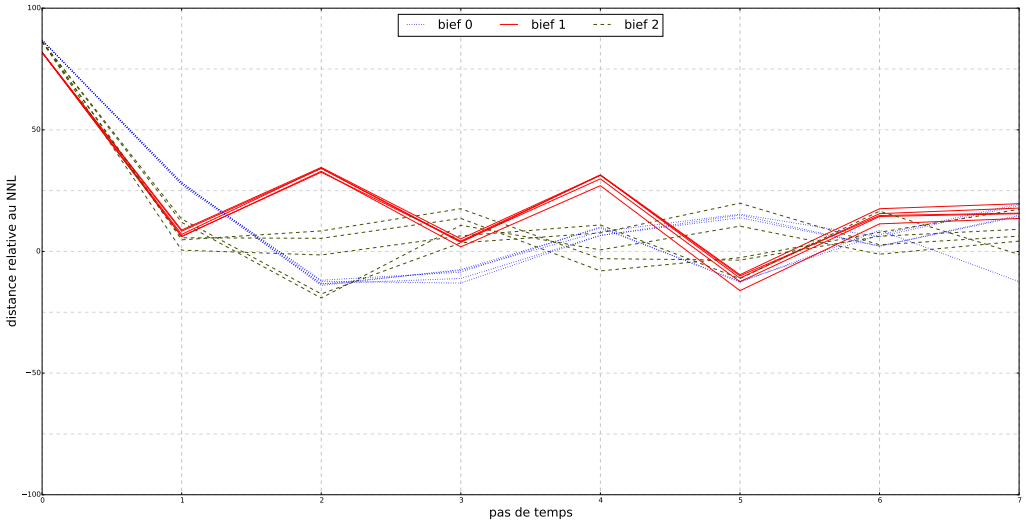

FIGURE 5.7 – États initiaux proches du HNL

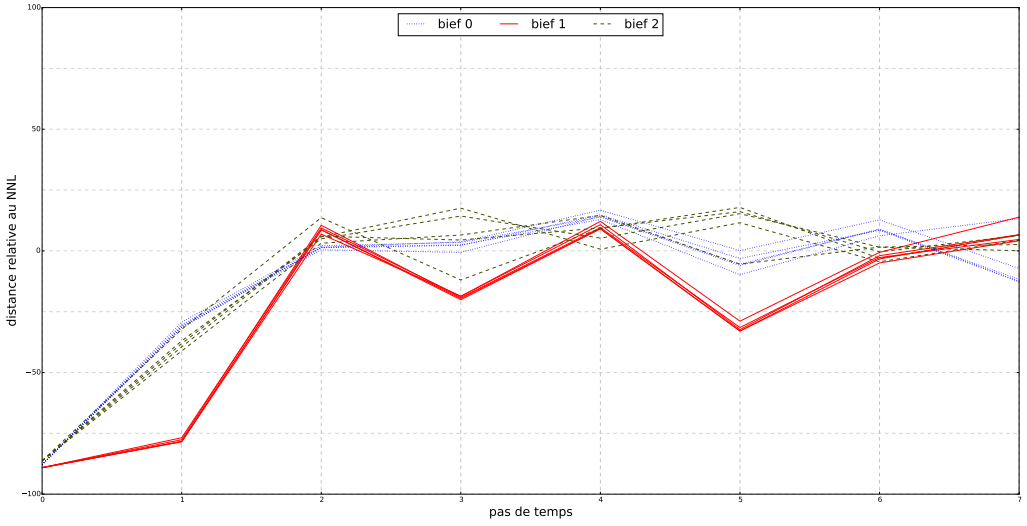

FIGURE 5.8 – États initiaux proches du LNL

Les deux dernières conditions d'évolution du réseau voient un trafic plus important de 10%, sur la figure 5.9, et moins important de 10%, sur la figure 5.10, que prévu. Dans les deux cas, les biefs 0 et 2 restent relativement proches de leur NNL respectif. Cependant, le bief 1 montre des difficultés à gérer ses volumes d'eau. Cela est principalement dû aux contraintes inhérentes à ce bief. Ce bief possède en effet la plus grande différence entre les volumes entrants et sortants, ce qui le rend très sensibles aux variations de trafic.

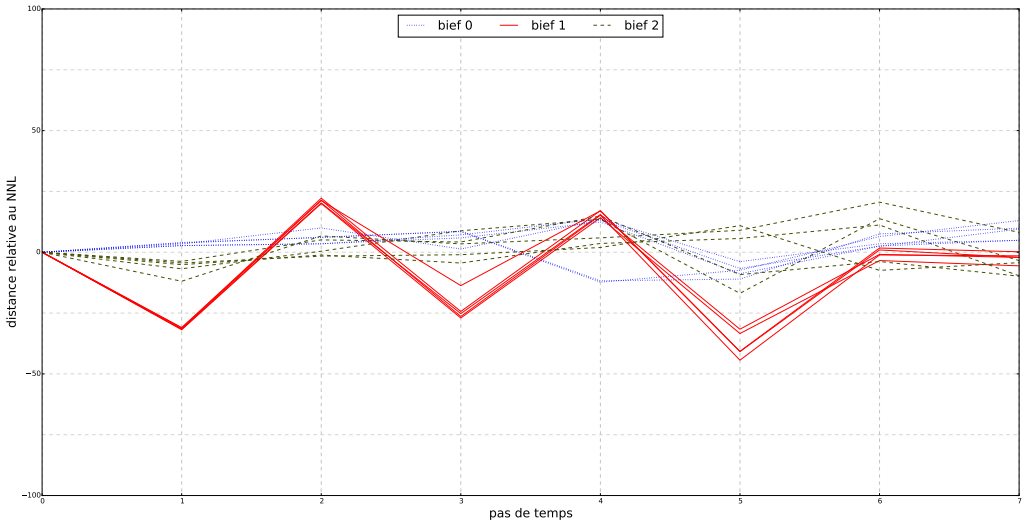

FIGURE 5.9 – Trafic 10% plus important que prévu

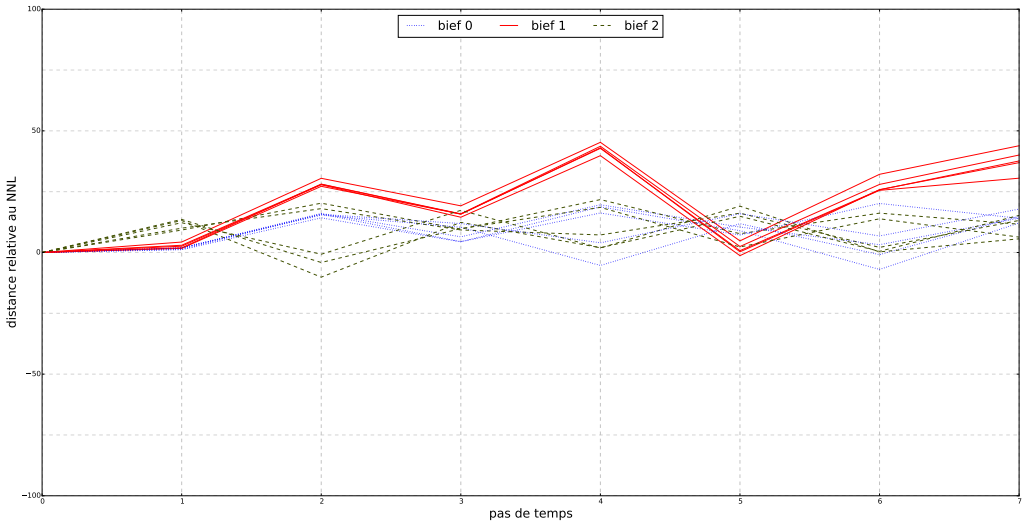

FIGURE 5.10 – Trafic 10% moins important que prévu

5.2.2 Scénario 2 : conditions d'étiage

Durant les périodes estivales ou de sécheresse, les débits des rivières non canalisées (via des points de transfert non contrôlés) se jetant dans le réseau sont réduits. L'« étiage » est le nom de la période où les cours d'eau sont à leurs niveaux minimums. Le trafic ne changeant pas durant cette période, il sera donc nécessaire de maintenir les conditions de navigation à un niveau acceptable malgré des conditions de gestion détériorées. La politique jointe a été obtenue en 19 minutes de calcul. Un tel scénario a été simulé avec les valeurs de volumes des points de transfert non contrôlés fournis dans la table 5.7. Ces valeurs reprennent les estimations de réduction du débit du projet Explore2070 [Chauveau et al., 2013].

Point de transfert	1	2	8	10
Volumes (m^3)	108 000	−12 960	15 120	40 608
Débits (m^3/s)	2,50	−0,30	0,35	0,94

TABLE 5.7 – Volumes déplacés par les points de transfert incontrôlables sur une période de 12 heures durant les périodes d'étiage

Dans un contexte d'étiage et avec des niveaux initiaux proches du NNL, voir la figure 5.11, maintenir le niveau idéal des biefs est toujours possible, bien que plus difficile. À cause des réductions des déplacements incontrôlables, le bief 1 ne peut plus maintenir son niveau au NNL durant les périodes de jour ni le compenser durant la nuit. Son bief amont possède assez de surplus de ressource durant la journée pour le compenser, mais la coordination entre les agents étant sous-optimale, ce surplus est transféré vers l'extérieur du réseau, empêchant ainsi le maintien le plus optimal des conditions de navigation pour le bief 1. Malgré tout, le rectangle de navigation est respecté et l'écart maximum d'un bief à son NNL est de 30% (1,5 cm).

Utilisant la même politique que la simulation précédente, la simulation ayant comme conditions initiales des niveaux proches du HNL (voir la figure 5.12) est donc affectée par les mêmes oscillations. Néanmoins, le système retrouve rapidement les niveaux idéaux des biefs. Ceci est cohérent avec la réduction des volumes entrant durant les conditions d'étiage. À l'inverse, lorsque les biefs sont initialement très proches de leur LNL (voir la figure 5.13), il n'est pas possible de maintenir le rectangle de navigation des biefs 0 et 1 de façon simultanée, en particulier en début de simulation. De ce fait, les agents se sont coordonnés pour ne rétablir directement qu'un seul bief. Il n'y a pas de priorité sur les biefs, les biefs les moins contraints se trouvent alors rétablis en premier. La récupération du bief 0 durant la première période permet une récupération plus rapide du bief 2. Seul le bief 1 violera les conditions de navigation, en atteignant un écart relatif de −240% au NNL, pour permettre aux deux autres biefs de retrouver rapidement des conditions idéales de navigation. Les conditions de navigation du bief 1 seront restaurées durant la deuxième moitié de la simulation, mais le réseau n'aura pas atteint des conditions idéales de navigation avant la fin de la période considérée.

Les biefs étant déjà fortement contraints par la navigation, une augmentation non anticipée

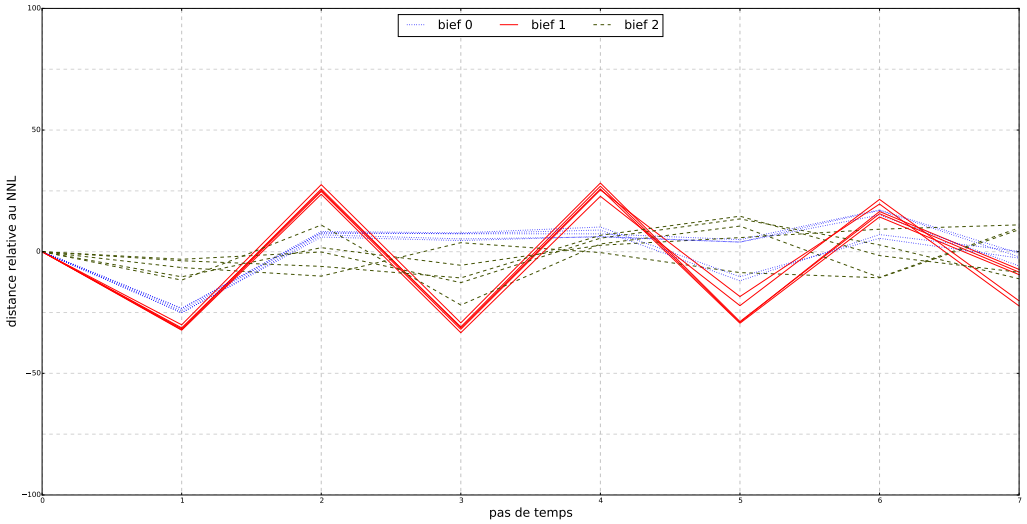

FIGURE 5.11 – Gestion du réseau sous conditions d'étiage avec les NNL comme états initiaux, évolution du volume des biefs relatif à leur rectangle de navigation avec un pas de temps de 12 heures

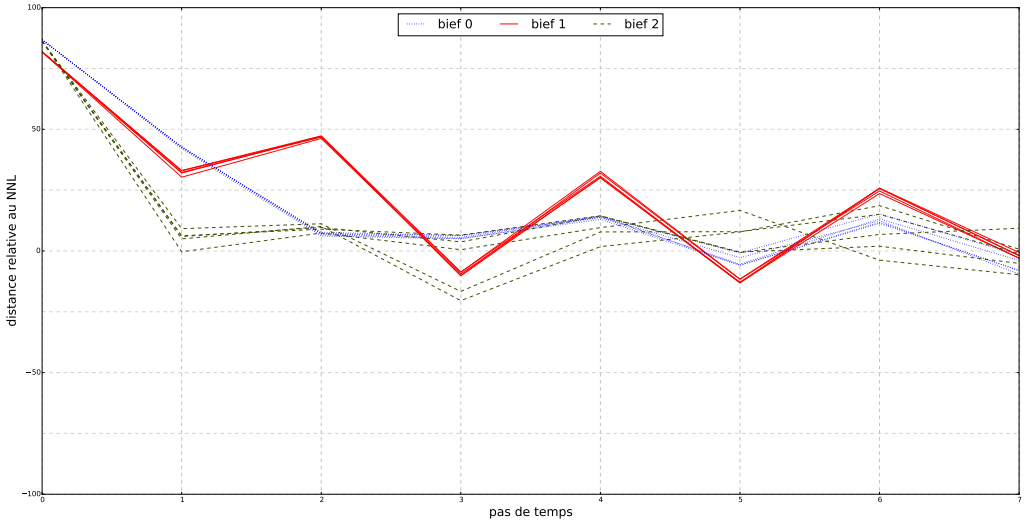

FIGURE 5.12 – États initiaux proches du HNL durant l'étiage

de celle-ci implique une plus grande intensité des oscillations du niveau des biefs, voir figure 5.14. Inversement, lorsque le trafic est réduit, voir la figure 5.15, l'amplitude des oscillations diminue, notamment celles du bief 1. La politique anticipe un plus grand trafic que celui simulé et le bief 1 est surcontraint par la navigation. Les variations entre les périodes de jours et de nuits sont donc naturellement plus faibles.

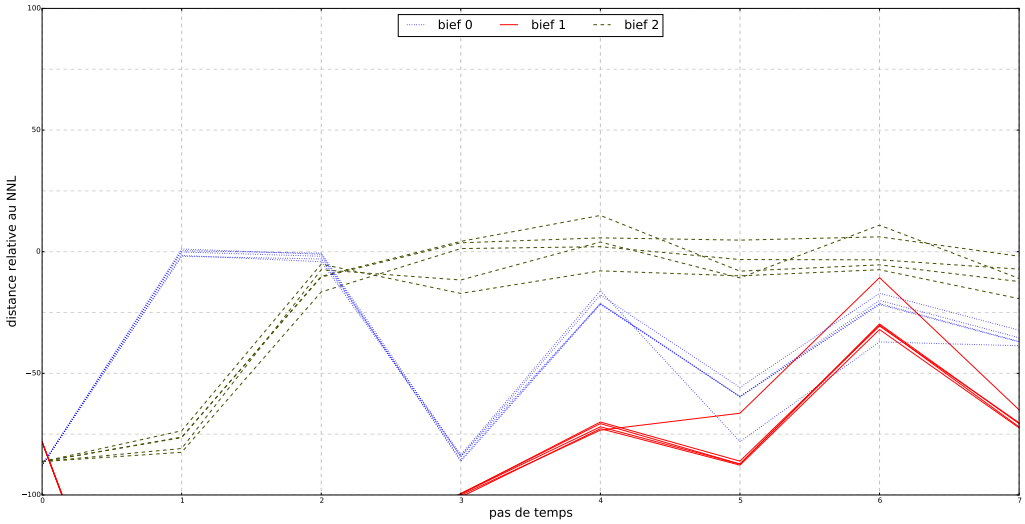

FIGURE 5.13 – États initiaux proches du LNL durant l'étiage

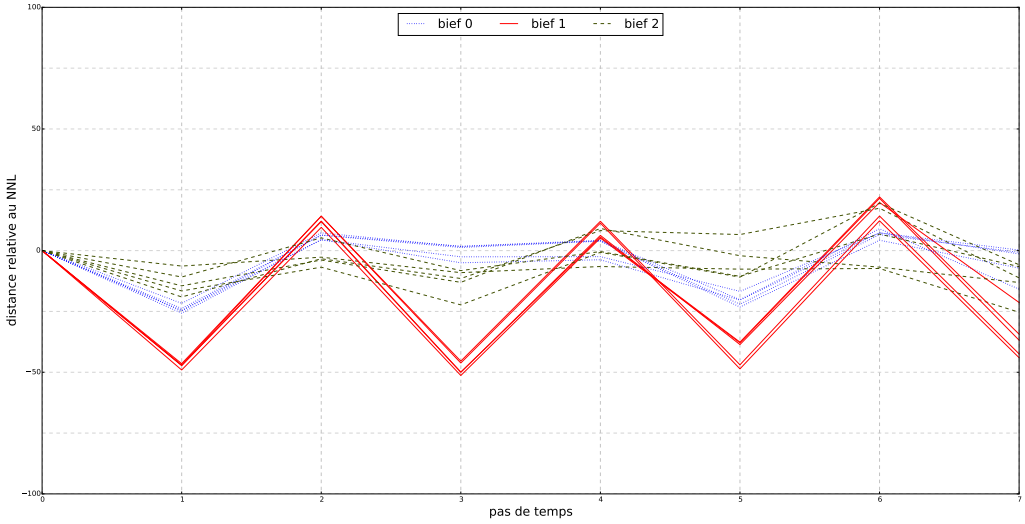

FIGURE 5.14 – Trafic 10% plus important que prévu durant l'étiage

Dans ces conditions de gestion plus contraignantes, les capacités d'optimisation de l'approche proposée sont limitées. L'absence d'optimalité de la politique jointe conduit à quelques défauts de coordination entre les agents. Néanmoins, la capacité à sacrifier une partie du réseau pour améliorer l'intégrité de celui-ci est aussi visible sur la figure 5.13. Bien que les conditions de simulations utilisées soient extrêmes, avec jusqu'à 70% de réduction des débits incontrôlables, l'approche fournit une gestion valide du réseau.

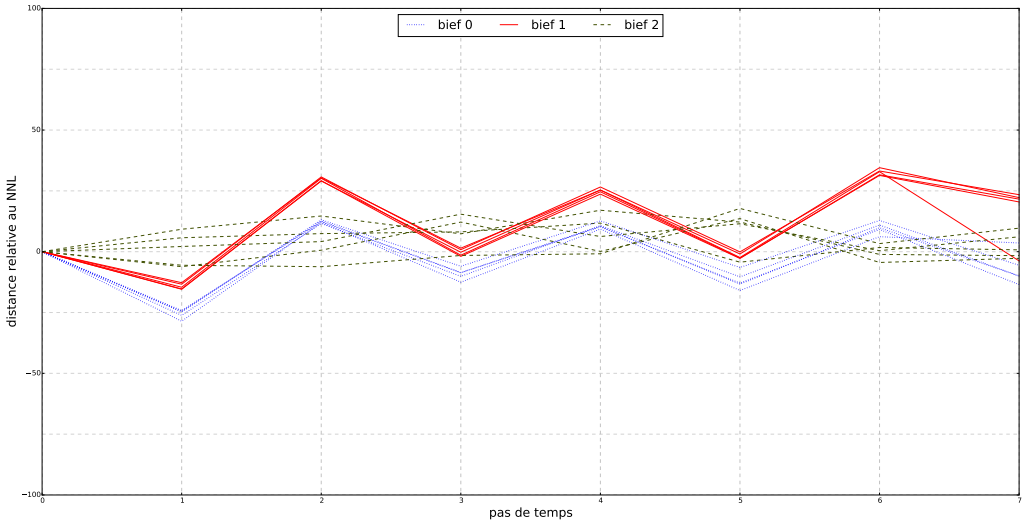

FIGURE 5.15 – Trafic 10% moins important que prévu durant l'étiage

5.2.3 Scénario 3 : conditions futures

Dans un futur proche, une ouverture de la navigation est prévue sur les périodes de jour et de nuit. Cette augmentation des horaires de navigation est prévue pour soutenir une augmentation du trafic fluvial. Ces conditions ont été appliquées au réseau considéré afin de tester sa capacité à supporter une augmentation du trafic malgré la suppression des périodes de repos la nuit.

Ce scénario utilise les mêmes volumes incontrôlables que lors de conditions normales. Par simplicité de modélisation, l'hypothèse est faite que le trafic nocturne est équivalent au trafic journalier, ce qui correspond ainsi une augmentation de 100% de l'utilisation des écluses chaque jour. L'obtention de la politique jointe pour ces conditions de navigation a nécessité 30 minutes.

Lorsque les biefs se trouvent dans des conditions initiales idéales, voir la figure 5.16, la politique obtenue est globalement capable de maintenir les biefs proches de leur NNL sur la durée de la simulation. Puisque la navigation est autorisée la nuit, les conditions de navigation sont les mêmes à chaque pas de temps. La présence de la navigation la nuit rend le bief 1 entièrement dépendant des transferts du bief 0. Il est possible de remarquer que les oscillations du volume du bief 1 sont décalées d'une période. Dans ce scénario, les simulations commencent et se terminent par une hausse du niveau du bief 1, à l'inverse des deux premiers scénarios (voir les figures 5.5 et 5.11).

La récupération des biefs proches de leur HNL (voir la figure 5.17) est plus efficace que dans des conditions normales. La suppression des périodes de remplissage la nuit semble faciliter l'écoulement du surplus d'eau. Le rétablissement des biefs depuis leur LNL (voir la figure 5.18) est plus lent avec les conditions futures, car l'augmentation du trafic réduit les quantités d'eau disponibles pouvant être déplacées à chaque instant. Cette résolution restaure les trois biefs

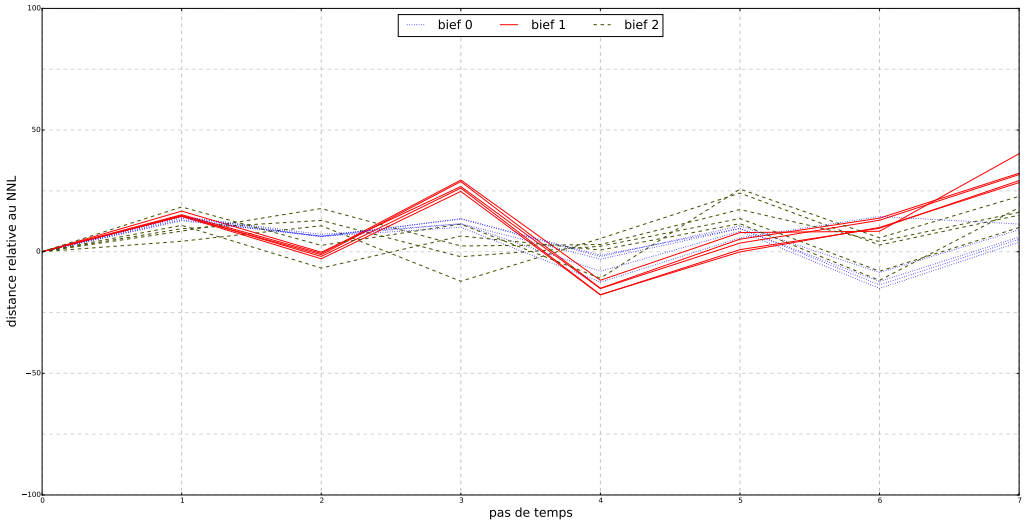

FIGURE 5.16 – Gestion du réseau sous conditions futures d'augmentation du trafic avec les NNL comme états initiaux, évolution du volume des biefs relatif à leur rectangle de navigation avec un pas de temps de 12 heures

de façon simultanée, ce qui est préférable pour la fonction de coût du modèle. Celle-ci étant quadratique, avoir tous les biefs à une distance moyenne de leur NNL induira un coût moins important que d'avoir un seul bief très éloigné et les autres très proches.

Si le trafic est plus faible que prévu, voir la figure 5.20, la politique jointe précédemment calculée reste efficace. Le trafic étant moins important que prévu, le bief 1 se remplira plus que prévu et renforcera sa déviation à la fin de la simulation. Dans le cas inverse, sur la figure 5.19, le niveau du bief 1 reste constant (25%) jusqu'à la fin, où il monte de façon analogue aux autres simulations de cette politique. Cela peut s'expliquer par le fait qu'à chaque instant l'écart au NNL est compensé par son bief amont. Cependant, comme la réduction de la demande de navigation affecte tous les pas de temps, la compensation est annulée. La politique obtenue fournit une solution valide en cas de changement imprévu. Néanmoins si les changements affectent l'intégralité de l'horizon de simulation, il pourrait être préférable de calculer une nouvelle politique en ajustant les prévisions des conditions futures.

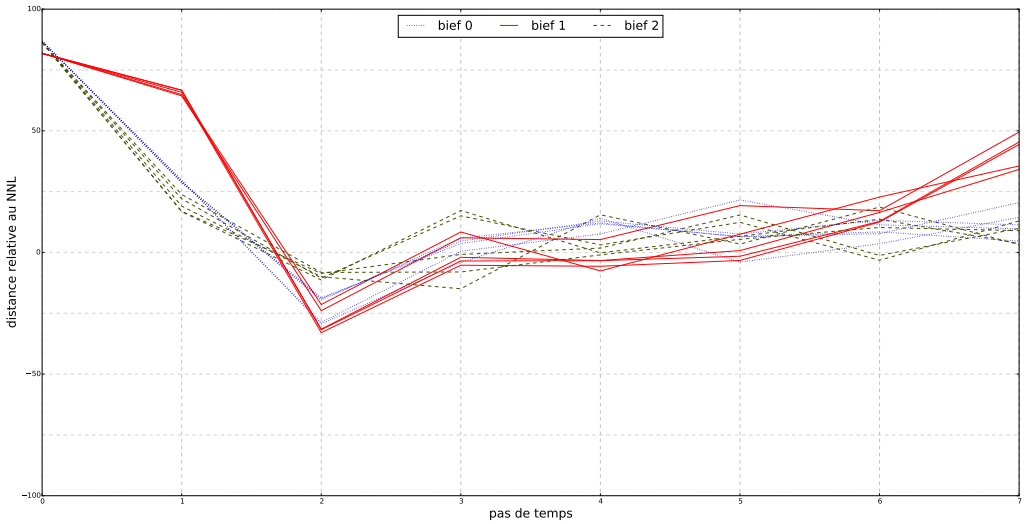

FIGURE 5.17 – États initiaux proches du HNL selon des conditions futures

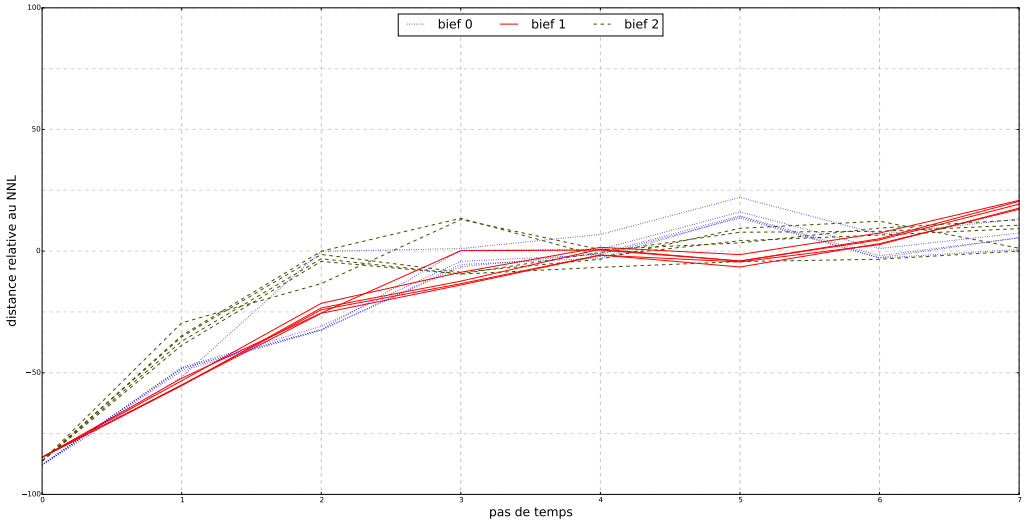

FIGURE 5.18 – États initiaux proches du LNL selon des conditions futures

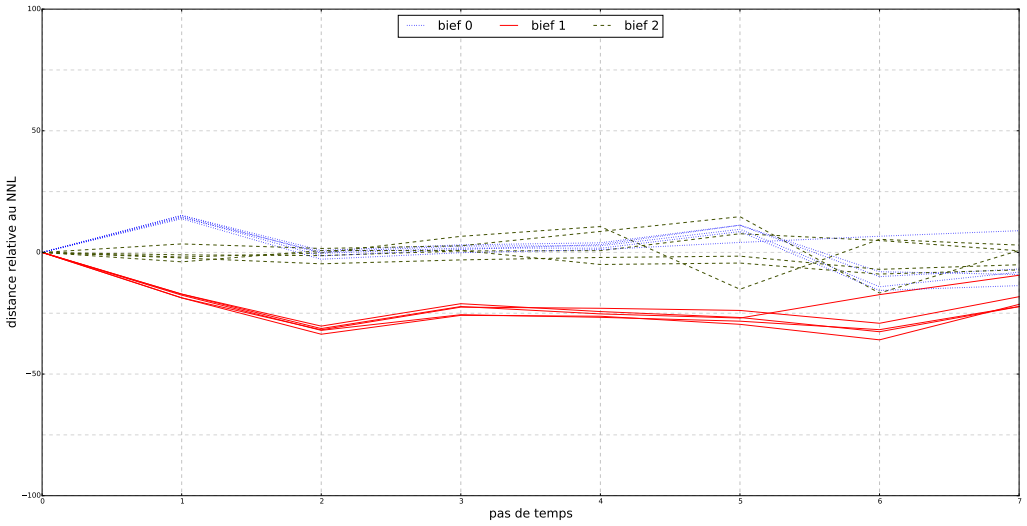

FIGURE 5.19 – Trafic 10% plus important que prévu selon des conditions futures

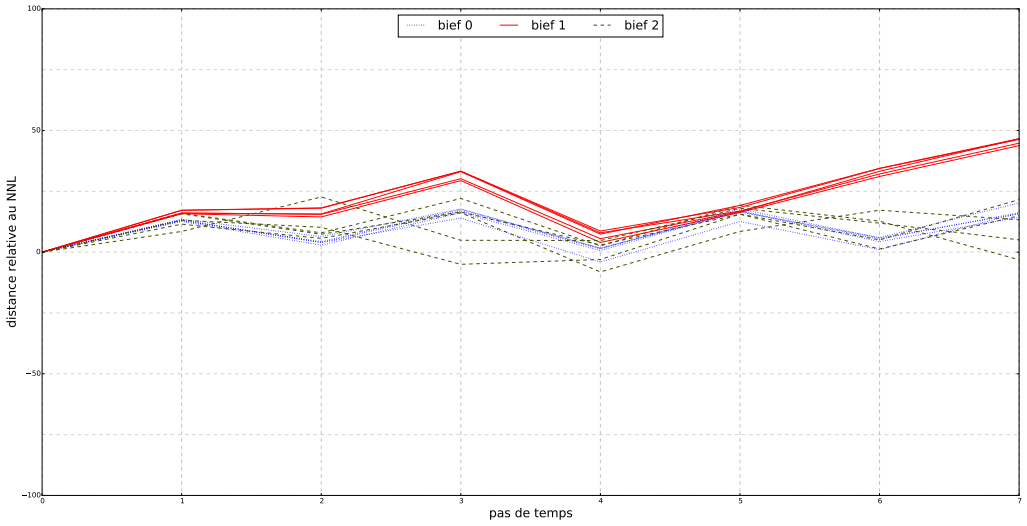

FIGURE 5.20 – Trafic 10% moins important que prévu selon des conditions futures

5.2.4 Scénario 4 : conditions d'un événement pluvieux

Ce scénario reprend les conditions actuelles normales de navigation et y ajoute la connaissance d'un événement probable de pluie localisée. Cette perturbation impliquerait une arrivée de $700\,000\,\mathrm{m}^3$ d'eau sur le bief 1, en débit non contrôlé. Ce volume a été choisi pour que la pluie produise une dégradation importante du niveau du bief lorsqu'il se trouve proche de son NNL, mais sans le faire sortir du rectangle de navigation. Ainsi, l'anticipation n'est pas nécessaire pour éviter la sortie du rectangle de navigation, bien qu'elle reste la solution préférable. La période de pluie a une probabilité de se produire de 0,60 durant la quatrième période de simulation. La politique jointe a été trouvée en 33 minutes. Pour ce scénario, les différences de simulations entre la politique ne connaissant pas l'existence d'une perturbation et celle la prenant en compte seront mises en avant.

La politique calculée ayant connaissance la pluie, voir la figure 5.21, est capable d'anticiper cet événement en baissant le niveau du bief 1 lors de la troisième période. Cet abaissement permet d'éviter une dégradation trop importante du bief par la pluie, mais aussi lorsque celle-ci ne se produit pas. La coordination entre les agents permet par la suite de retrouver un niveau proche de l'idéal. En comparant avec la politique obtenue sans prise en compte des incertitudes (scénario 1) et donc de la pluie (figure 5.22), cette anticipation améliore significativement le niveau d'eau. L'anticipation totale de la pluie n'étant pas possible sans dégrader les conditions de navigation, la politique calculée propose une des meilleures anticipations possibles. En effet, l'anticipation de cette perturbation induit un écart au NNL du bief 1 de -43% ce qui est équivalent à l'écart de $+50\%$ lors de l'occurrence de la perturbation (figure 5.21). Une anticipation plus importante conduirait à une dégradation de la solution, notamment dans les cas où la pluie n'affecte pas le réseau.

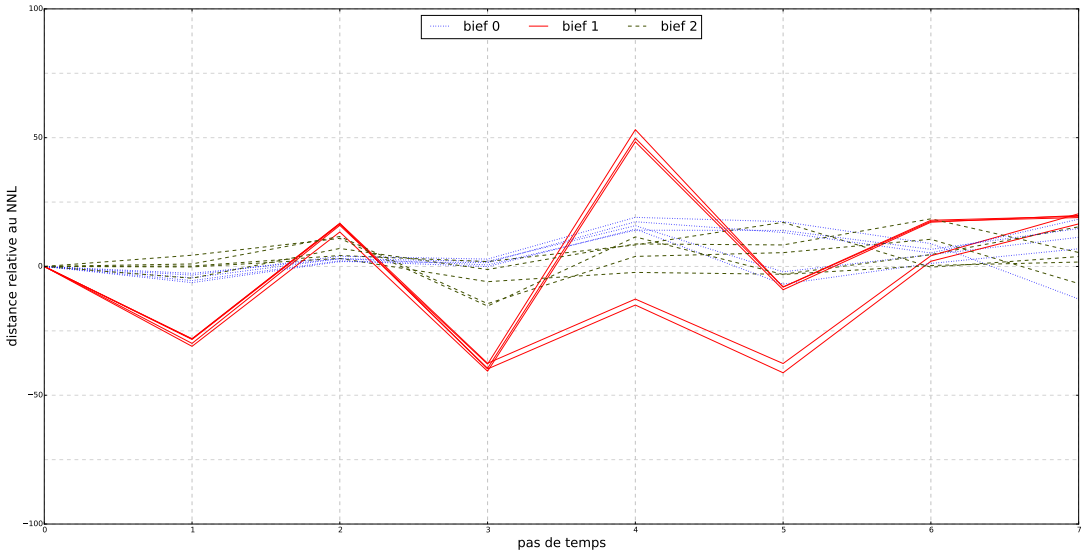

FIGURE 5.21 – Pluie possible prévue et anticipée sur le bief 1, évolution du volume des biefs relatif à leur rectangle de navigation avec un pas de temps de 12 heures

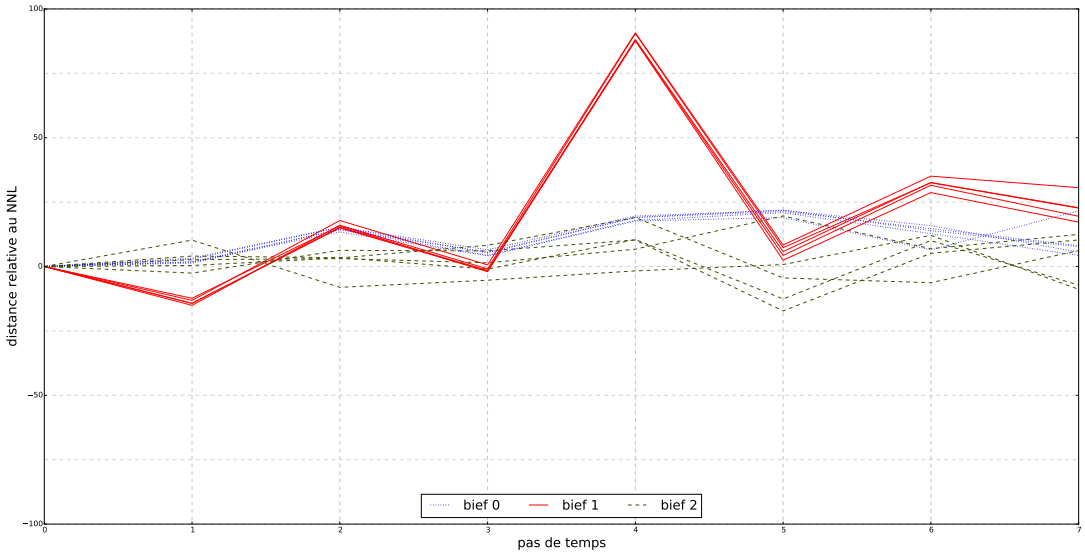

FIGURE 5.22 – Pluie non prévue sur le bief 1

5.2.5 Prise en compte des incertitudes sur les transferts incontrôlés

Une seconde version du premier scénario a été réalisée en prenant en compte un grand nombre d'incertitudes sur le trafic et les volumes incontrôlés en déplacement sur le réseau. Le but étant d'observer l'impact de la prise en compte, ou non, des incertitudes lors de la planification.

Le nombre de bassinées (ou éclusées) moyen par écluse de ce scénario est identique à celle de la table 5.3, mais cette valeur n'est qu'une estimation du nombre de bassinées pour chaque période de navigation. Les variations de l'utilisation des écluses sont décrites dans la table 5.8. Ces variations sont les mêmes pour toutes les écluses et pour chaque période de navigation. Elles vont d'une variation maximale de volumes d'eau déplacés de ±10% pour l'écluse 0 à ±20% pour l'écluse 7 (cf. figure 5.4 page 96). Le débit des rivières connectées au réseau est aussi variable et défini dans la table 5.9. Les variations sur les bassinées et les rivières sont considérées comme indépendantes les unes des autres.

Variation (bassinée)	-2	-1	0	+1	+2
Probabilité	7%	24%	38%	24%	7%

TABLE 5.8 – Incertitudes sur le nombre de bassinées de chaque écluse

Variation (m^3/s)	-0,2	-0,1	0	+0,1	+0,2
Variation (m^3)	-8640	-4320	0	+4320	+8640
Probabilité	5%	16%	58%	16%	5%

TABLE 5.9 – Incertitudes sur les entrées non contrôlées 1,8 et 10 (cf figure 5.4)

Une politique jointe prenant en compte l'ensemble de ces incertitudes a été calculée. L'obtention de cette politique fut lente, en nécessitant environ 27 jours de calcul. La majorité du temps de calcul fut utilisée pour la construction des fonctions de transition de chaque agent. Ce temps de calcul sera discuté plus en détail dans la section 5.4.1.

La première expérimentation consiste en une comparaison de la politique jointe considérant les incertitudes avec une politique jointe ne les considérant pas (scénario 1). Les deux politiques sont comparées sur 500 000 simulations pour observer l'effet des politiques sur un grand nombre de combinaisons possibles de variations. Les critères de comparaison sont le nombre moyen de biefs en dehors de leur rectangle de navigation sur la durée de simulation, la distance relative moyenne au NNL (0% = NNL et 100% = HNL ou LNL), l'écart-type des distances relatives au NNL et la distance maximale au NNL observé sur une simulation. Les résultats sont donnés dans la table 5.10.

La prise en compte des incertitudes permet une gestion plus tolérante des volumes d'eau qui permet de limiter les écarts au NNL lorsque des variations du nombre de bassinées ou de débit des rivières se produisent. Ignorer ces incertitudes fait courir le risque de violer les conditions de navigation d'un ou plusieurs biefs sur la durée de la simulation et résulte en une gestion plus incertaine avec de plus grandes variations d'écart au NNL.

Une seconde expérimentation a été réalisée afin d'évaluer la politique prenant en compte les variations possibles dans un scénario avec aucune variation par rapport à ce qui a été prévu.

	Utilisation des incertitudes	Incertitudes ignorées
Moyenne de biefs hors du rectangle de navigation	0,000	0,001
Distance au NNL	10,396%	13,086%
Écart-type des distances au NNL	9,270%	12,527%
Distance maximale	85,56%	110,700%

TABLE 5.10 – Statistiques des différents scénarios avec variations incertaines

Cela correspond à appliquer la politique sur le scénario 1, présenté en section 5.2.1 (page 97). Les résultats des 500 000 simulations sont visibles sur la table 5.11.

	Utilisation des incertitudes	Incertitudes ignorées
Moyenne de biefs hors du rectangle de navigation	0,000	0,000
Distance au NNL	8,368%	8,494%
Écart-type des distances au NNL	6,945%	7,101%
Distance maximale	30,991%	31,113%

TABLE 5.11 – Statistiques des différents scénarios sans variations incertaines

Malgré l'absence de variations durant la simulation, la politique les prenant en compte produit des résultats équivalents, voire un peu meilleurs à celle construite pour ce scénario. En moyenne, l'impact des variations est négligeable, celles-ci possédant autant de chance d'être une augmentation qu'une réduction de volume. Ainsi, il est cohérent que les deux approches obtiennent des résultats similaires sur cet exemple. Les différences en faveur de la politique prenant en compte des incertitudes sont liées à une meilleure coordination entre les agents. Le risque inhérent aux incertitudes de dégrader la solution dans les deux sens (augmentation ou réduction des volumes d'eau), pénalise plus fortement l'écart au NNL. Ceci est visible par une réduction des oscillations sur la figure 5.23 en comparaison de la figure 5.5.

Bien que fournissant de meilleurs résultats dans les deux situations, la prise en compte des incertitudes n'est pas forcément préférable à cause de l'augmentation significative du temps de calcul. Cependant, dans cet exemple les écarts d'évaluations sont particulièrement faibles principalement puisque l'espérance des variations sur chaque point de transfert est toujours de

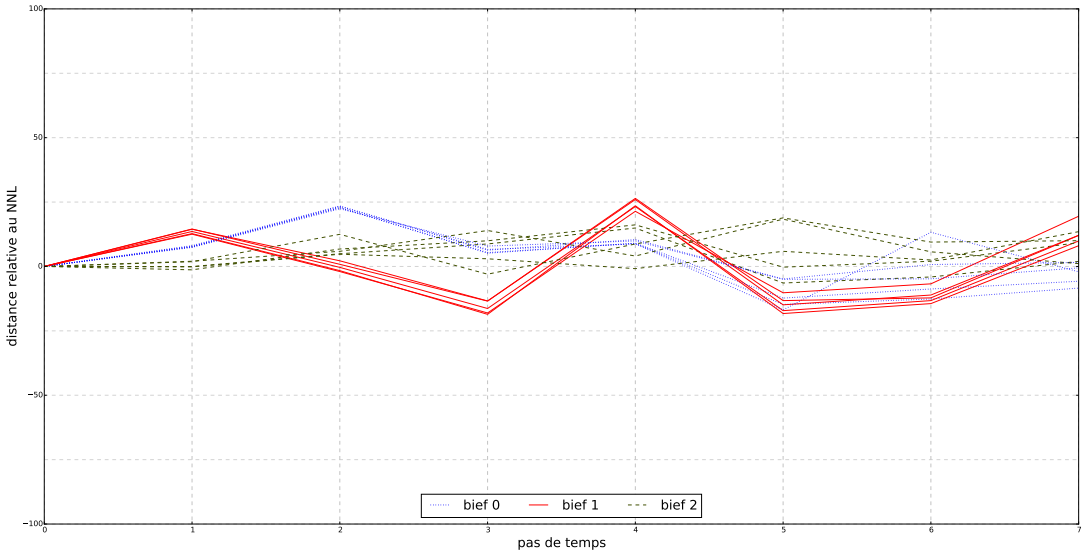

FIGURE 5.23 – Gestion du réseau avec une politique prenant un compte des incertitudes sur les échanges non contrôlés dans des conditions normales, évolution du volume des biefs relatif à leur rectangle de navigation avec un pas de temps de 12 heures

0. Une optimisation de l'implémentation de l'algorithme pourrait réduire les écarts de temps de calcul. Par exemple, en ne reconstruisant pas la fonction de transition entièrement à chaque itération, mais en ne prenant uniquement en compte que les choix estimés du voisinage

5.3 Application sur un réseau réel de 7 biefs

Une autre expérimentation a été réalisée sur un réseau réaliste des voies navigables des Hauts-de-France de plus grande taille. Les données fournies à notre disposition sur ces biefs sont limitées quant aux tailles des biefs et aux volumes contrôlés et non contrôlés les affectant. De ce fait, un sous-réseau composé de 7 biefs est considéré (figure 5.24). Le réseau est schématisé sur la figure 5.25, où les biefs sont représentés par les nœuds du graphe, les points de transfert contrôlables par des arcs pleins, les points de transferts non contrôlables par des arcs en pointillés. Les points de transferts ont été répartis entre 9 agents représentés par des ovales grisés. Le nombre d'agents a été défini de façon à ce que chaque agent ne possède pas plus de deux points de transfert contrôlables (arcs pleins) ni n'observe plus de deux biefs, ceci afin de limiter la taille des fonctions de transition de chaque agent.

Les caractéristiques du réseau sont présentées dans la table 5.12. Elle répertorie les volumes correspondant aux rectangles de navigation des biefs modélisés, au NNL ainsi que le nombre d'intervalles utilisés pour la discrétisation. La discrétisation des biefs en intervalles est liée à celle des points de transfert. Le but de cette discrétisation étant de limiter le nombre d'états pouvant

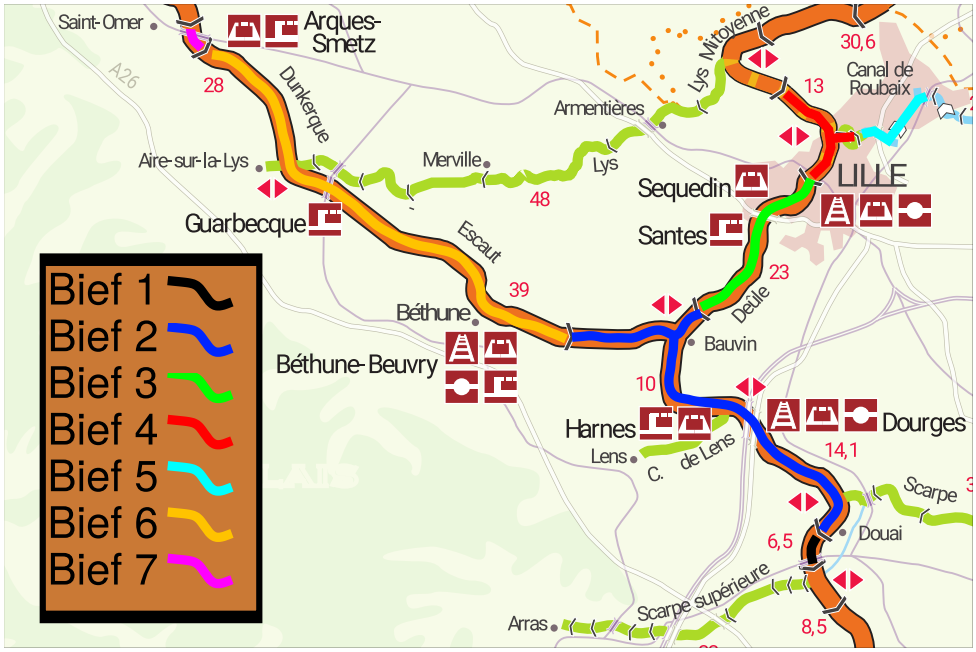

FIGURE 5.24 – Position du réseau de 7 biefs sur une carte du Nord-Pas-de-Calais

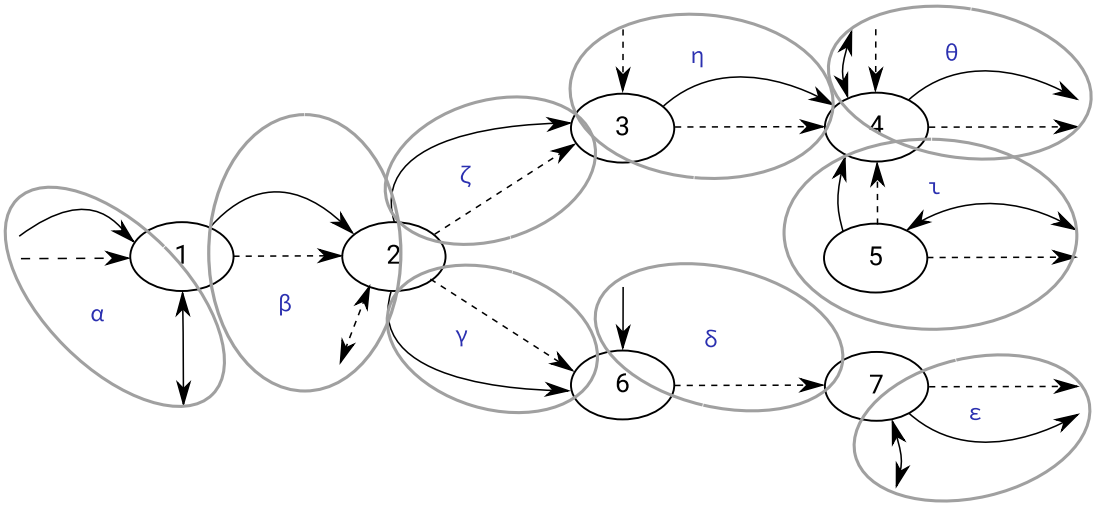

FIGURE 5.25 – Sous-réseau des voies navigables du nord de France, 7 biefs (nœuds), 26 points de transferts (arcs) et 9 agents (ovales gris)

être atteints en appliquant une action jointe sur un bief. Elle utilise la discrétisation décrite en section 3.4.2 (page 63). Une simplification du modèle a été effectuée dans le but de réduire le nombre d'actions par agent (table 5.13) : les points de transfert contrôlables connectant deux mêmes biefs sont fusionnés en additionnant les bornes de leurs intervalles. Ceci permet ainsi une

Bief	Nom	LNL (m^3)	NNL (m^3)	HNL (m^3)	# intervalles
1	Courchelettes - Douai	407 699	414 120	420 560	13
2	Douai - Don - Cuinchy	8 654 381	8 772 934	9 010 040	12
3	Don - Grand Carré	3 766 098	3 824 038	3 881 978	13
4	Grand Carré - Quesnoy - Marquette	832 222	851 818	888 222	12
5	Marq - Marquette	185 831	190 221	194 611	11
6	Cuinchy - Fontinettes	9 348 300	9 458 280	9 568 260	17
7	Fontinettes - Flandres	112 320	114 102	115 885	9

TABLE 5.12 – Propriétés du réseau modélisé de 7 biefs du nord de France

Agent	α	β	γ	δ	ϵ	ζ	η	θ	ι
Actions	609	144	100	360	600	250	584	450	1 634

TABLE 5.13 – Nombre d'actions par agent

discrétisation plus fine avec moins de variabilité lors de l'application de la politique. Cela conduit à l'obtention d'un seul intervalle au lieu de deux. Cette optimisation est aussi possible dans le cas de connexions externes, car l'état des éléments non modélisés est ignoré.

La planification et la simulation se déroulent sur une période d'une demi-semaine, de 7 périodes de 12 heures. Ainsi les agents ont un maximum de $12 \times 17 \times (7 + 1) = 1632$ états, pour l'agent γ, et 1634 actions, pour l'agent ι. Une représentation mono-agent avec la même modélisation impliquerait environ 2×10^8 états et 2×10^{23} actions.

	Scénario 1	Scénario 2	Scénario 3 utilisant la politique du scénario 2	Scénario 3
Moyenne de biefs hors du rectangle de navigation	0,00	0,00	0,02	0,00
Distance au NNL	11,37%	8,31%	11,65%	10,94%
Écart-type des distances au NNL	11,58%	9,21%	14,34%	14,13%
Distance maximale	72,00%	61,66%	106,41%	83,61%

TABLE 5.14 – Statistiques des différents scénarios sur 500 000 simulations

La planification est effectuée sur trois scénarios différents :

1. situation normale : pas de trafic la nuit ;
2. situation future : autorisation du trafic de nuit, autant de trafic le jour et la nuit ;

3. situation future avec perturbation : semblable aux conditions précédentes, mais avec une perturbation qui a une probabilité d'affecter le bief 2.

Contrairement aux expérimentations sur un réseau de trois biefs, aucune expérimentation sur une période d'étiage n'a été effectuée faute d'informations quant à l'évolution des débits sur la majorité du réseau durant l'étiage. Le trafic est considéré comme constant sur les périodes de navigation. Les politiques jointes ont été obtenues par l'algorithme OCLP en 9 heures pour le premier scénario, 1 heure 30 pour le second et 2 heures 30 pour le troisième. La différence de temps entre le premier scénario et les deux derniers vient de la difficulté des agents à trouver un compromis. À cause de cette difficulté, la résolution tombe dans un cycle qui n'est détecté qu'après quelques heures puis brisé avant de terminer la résolution. Les indicateurs de performance de l'approche de chaque scénario ont été obtenus en utilisant 500 000 simulations, en prenant des valeurs aléatoires dans les intervalles choisis ; ils sont présentés dans la table 5.14. Une simulation déplaçant la valeur moyenne de chaque intervalle choisi est utilisée pour visualiser l'impact de la politique jointe dans des conditions idéales de navigation (figures 5.26, 5.27, 5.28, 5.29 et 5.30). Les biefs commencent à leur NNL et les perturbations pouvant affecter le réseau et leurs probabilités sont connues.

L'évolution du volume de chaque bief au cours du temps relativement à leur rectangle de navigation est visible sur les figures de 5.26 à 5.30. Les valeurs -100, 0 et 100 correspondent respectivement aux LNL, NNL et HNL de chaque bief.

La figure 5.26 rend visible l'utilisation de la politique jointe obtenue pour un scénario de gestion constitué uniquement de 12 heures de trafic par jour. Malgré un écart important durant la première période pour les biefs 4 et 6, et du bief 3 sur la dernière période, les biefs arrivent à maintenir un écart relatif au NNL favorable sur la durée. Ceci peut être aussi observé sur la table 5.14. L'écart moyen d'un bief à son NNL est de seulement 11,37% avec un écart-type de 11,58%. L'écart-type permet de donner un aperçu de l'impact du choix des volumes à transférer dans les intervalles d'action sélectionnés. Des variations importantes dans les transferts se produiront lorsque le volume observé ou idéal se trouve proche d'une borne d'un intervalle d'état. Dans de rares conditions un mauvais choix de volumes successifs dans les intervalles peut éloigner drastiquement un bief de son niveau normal, jusqu'à 72% d'écart au NNL (table 5.14). Cependant, cette déviation sera corrigée rapidement *a posteriori*.

Dans le contexte du trafic continu du second scénario, il est possible d'observer sur les courbes de la figure 5.27 et dans la table 5.14, que la politique jointe obtenue produit de meilleurs résultats sur les critères observés : écart au NNL et distance maximale. Cette différence est vraisemblablement liée à la disparition de la période de repos la nuit, qui simplifie les contraintes de certains biefs en permettant de réduire les échanges contrôlés avec le réseau extérieur. Une autre hypothèse est qu'avec un trafic continu et constant dans le temps, le système deviendrait plus simple à optimiser en cas d'absence de perturbation.

Similairement au quatrième scénario du réseau précédent (section 5.2.4 page 109), un événement de pluie est localisé sur le bief 2 avec 50% de chance d'occurrence d'apporter une grande

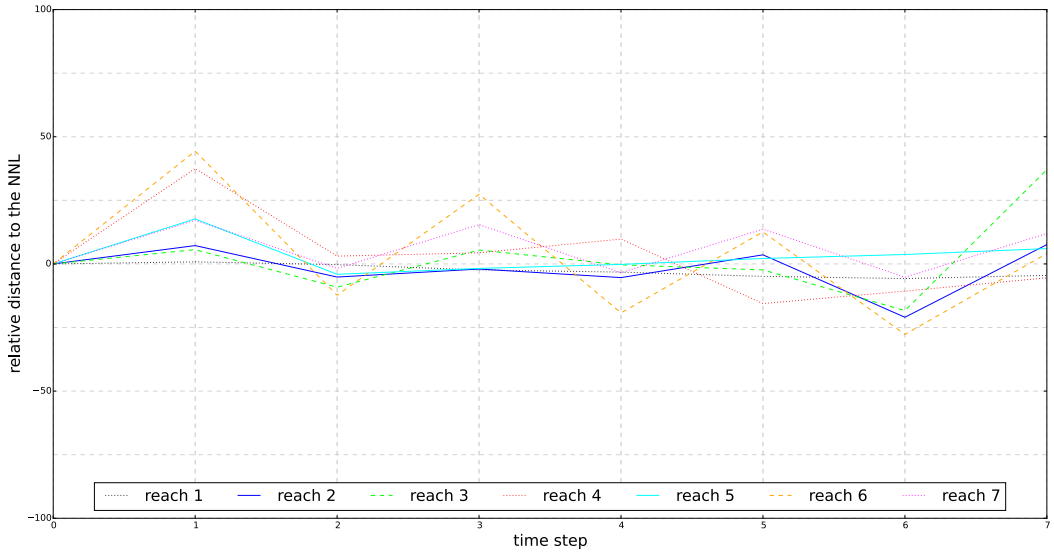

FIGURE 5.26 – Premier scénario : 12 heures de trafic par jour, biefs initialement au NNL, évolution du volume des biefs relatif à leur rectangle de navigation avec un pas de temps de 12 heures

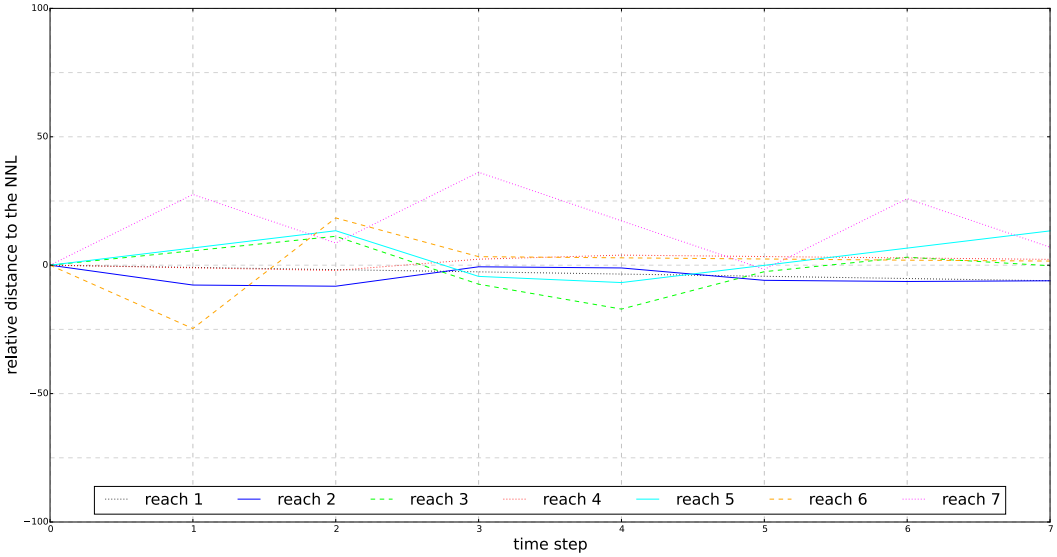

FIGURE 5.27 – Second scénario : 24 heures de trafic par jour, biefs initialement au NNL

quantité d'eau (230 000 m^3). Cet événement est suffisamment extrême pour faire approcher le bief de son HNL lorsqu'il se trouve à son NNL. Il n'est pas nécessaire d'anticiper cet événement pour rester dans le rectangle de navigation. Ceci est visible sur la figure 5.28, lorsque la politique jointe obtenue pour le scénario ne prend pas en compte cet événement et le subit. Sans antici-

pation, le bief 2 sera très fortement dégradé et pourra même violer ses conditions de navigation dans un faible nombre de cas, voir la table 5.14. Le rétablissement du niveau est possible. Il y a récupération de l'événement sur les biefs en aval.

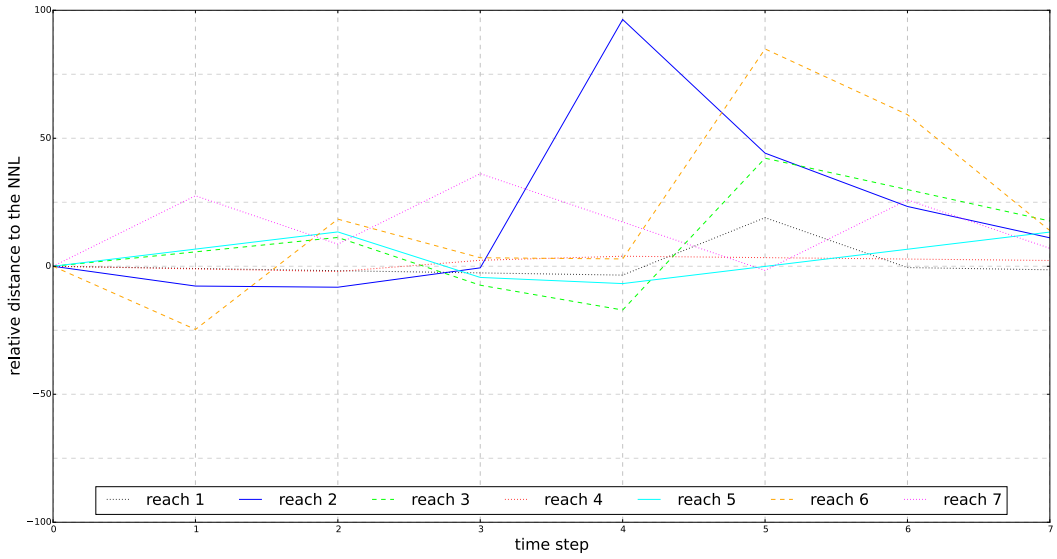

FIGURE 5.28 – Troisième scénario : pluie probable (50% de chance) sur le bief 2 utilisant la politique jointe obtenue pour le second scénario (pas de pluie prévue), biefs initialement au NNL

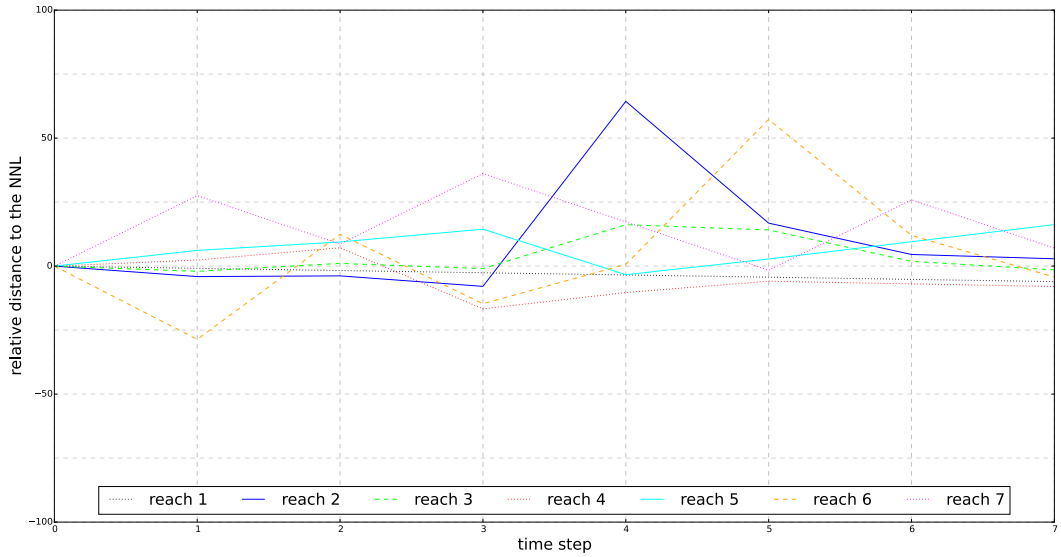

FIGURE 5.29 – Troisième scénario : pluie probable (50% de chance) sur le bief 2 utilisant la politique jointe obtenue pour ce scénario (pluie potentielle prévue), biefs initialement au NNL

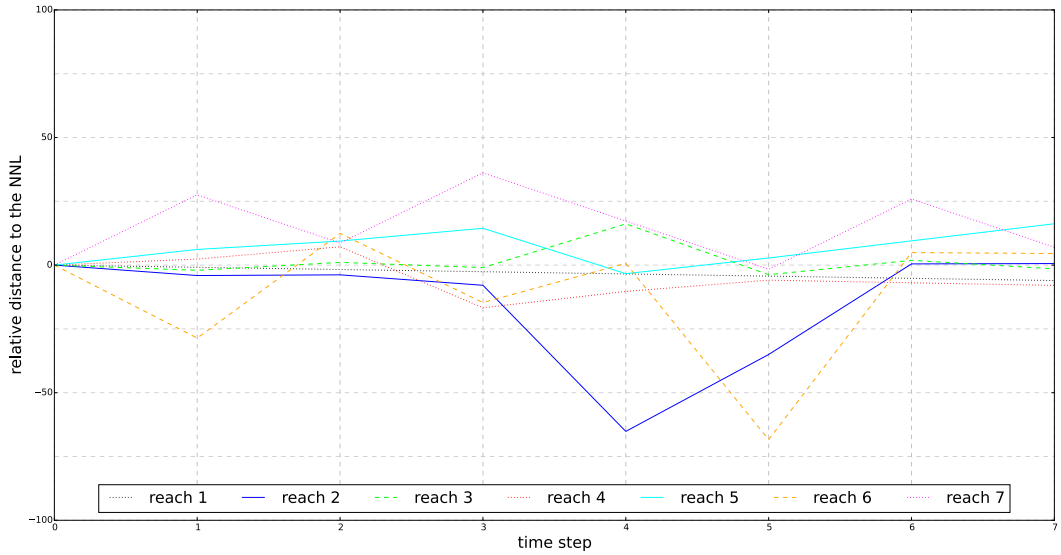

FIGURE 5.30 – Troisième scénario : 24 heures de trafic par jour anticipant un événement de pluie (50% de chance) affectant le bief 2, biefs initialement au NNL sans l'occurrence de la perturbation

Dans le cas où la perturbation peut être prévue, les agents sont capables d'anticiper la période de pluie afin de minimiser ses conséquences. De plus, l'événement n'étant pas certain, il faut aussi minimiser l'impact de l'anticipation. La figure 5.29 correspond à l'utilisation de la politique jointe lorsque l'événement se produit et la figure 5.30 dans le cas contraire. Lorsque la pluie est anticipée mais qu'il ne pleut pas, le bief 2 se vide dans les biefs aval sans les perturber, car ils s'attendent à cette arrivée d'eau. De cette façon, si la perturbation se produit, le bief se trouvera à distance plus raisonnable de son NNL. Le bief 6 est dégradé pour permettre au bief 2 de se rétablir, tout en restant dans son rectangle de navigation. Étant dépendant du bief amont pour maintenir son niveau, il sera forcément dégradé si celui-ci doit augmenter son volume. Cela lui conférera aussi la capacité à absorber un important volume d'eau provenant du bief 2.

Dans la table 5.14, la différence de valeurs pour l'écart moyen des biefs par rapport à leur NNL entre les approches avec et sans anticipation de l'événement pluvieux est faible. Ce faible écart à plusieurs causes. Premièrement, la gestion de la perturbation n'affecte que 3 biefs, dont un très légèrement, sur un maximum de 4 périodes. Dans un second temps, dans le cas de la politique sans anticipation, la perturbation n'arrive en moyenne qu'une fois sur deux et lorsqu'elle n'arrive pas la gestion du réseau est très efficace. Enfin pour limiter l'éloignement du bief lors de la perturbation, la politique avec anticipation dégrade le niveau du bief 2. De ce fait, même si la perturbation est bien mieux anticipée avec la troisième politique jointe, les écarts moyens des biefs à leur NNL et les écart-types correspondants restent proches.

En représentant un réseau plus important que précédemment, avec un plus grand nombre d'agents, l'approche OCLP est toujours capable de proposer des politiques de gestion efficace.

Dans les trois situations considérées, les simulations sont équivalentes à celles observées sur des réseaux de seulement trois biefs. De même, cette approche garde sa capacité d'anticipation des événements prévisibles et de récupération des événements imprévus.

5.4 Discussions

L'ensemble des expériences a mis en avant la capacité à contrôler un système de voies navigables. Une analyse fine des politiques a permis d'identifier les mécanismes d'anticipation et de recouvrement, qui sont appliqués automatiquement, sur un réseau particulier contenant une zone fortement contrainte. Ici, il est proposé de prendre plus de recul quant à l'approche introduite vis-à-vis du temps de calcul, d'une réduction de la discrétisation, des résultats et enfin de l'industrialisation de l'approche.

5.4.1 Temps de calcul

Plusieurs critères ont un impact significatif sur le temps de calcul nécessaire pour converger sur une politique jointe des agents :

1. la complexité des modèles locaux et donc de la décomposition ;
2. la topologie du réseau ;
3. la présence ou non de cycles durant la résolution.

Chaque itération lors de la résolution implique la construction d'un nouveau MDP puis sa résolution. Les cinq politiques calculées sur le réseau de trois biefs, dans la section 5.2 (page 94), ont nécessité un nombre d'itérations similaires pour converger vers une solution localement optimale. Cependant, la résolution prenant en compte un grand nombre d'incertitudes (section 5.2.5 page 110) a nécessité un temps de calcul bien plus long (27 jours contre 27 minutes). Cette augmentation de la durée de calcul est principalement due au calcul de la nouvelle fonction de transition. Pour prendre en compte les incertitudes dans la fonction de transition, les variations possibles affectant un agent devront être considérées dans le résultat de chaque action jointe à partir de chaque état. Lorsque la majorité des contrôleurs peut être affectée continuellement par des variations, le nombre de combinaisons d'actions (de l'agent), d'actions jointes (reçue du voisinage) et de variations devient pénalisant.

Comme les agents se trouvent synchronisés par l'échange de leur compteur et de leur politique adaptée, la vitesse de résolution sera définie par l'agent le plus lent. Ainsi, une décomposition en agent permettant de bien répartir les charges de modélisation, notamment des incertitudes, serait nécessaire pour limiter les différences de temps de calcul entre les agents et ainsi l'inactivité des agents les plus rapides. Par exemple, en utilisant un algorithme de partitionnement avec une heuristique visant à limiter la taille de la plus grande fonction de transition des agents.

La présence de cycles lors de la recherche d'une politique jointe peut significativement ralentir son obtention. Par exemple, le premier scénario du réseau à 7 biefs souffre de problème de cycles.

Sa résolution a nécessité 3 600 itérations avant que les cycles soient détectés puis brisés afin de permettre la terminaison. Par comparaison, les autres scénarios sur ce réseau n'ont requis qu'une centaine d'itérations.

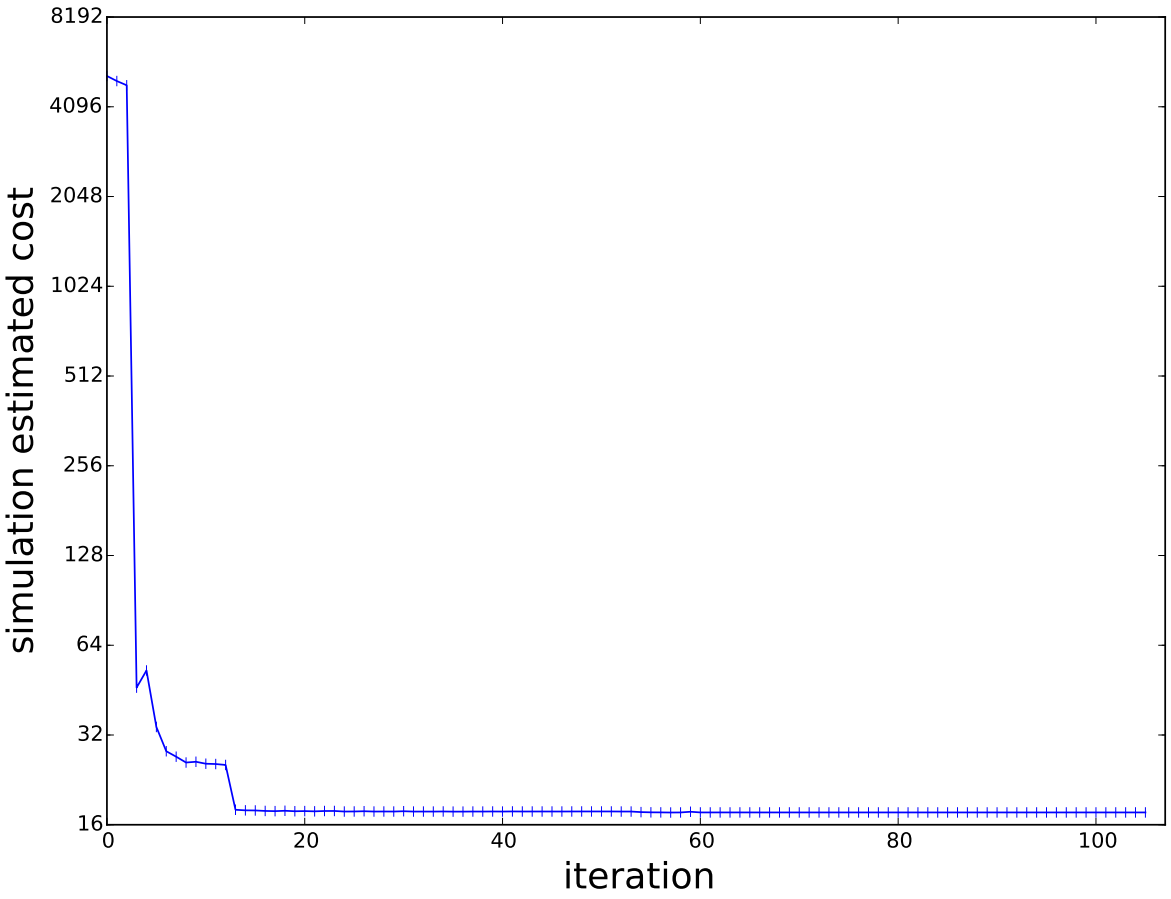

FIGURE 5.31 – Évolution du coût de la politique jointe au cours des cent premières itérations

Lorsque la résolution est trop longue, il est possible de se satisfaire d'une politique jointe avant la terminaison. En effet, sur les scénarios ici testés, les principales améliorations de la politique jointe sont situées sur les premières itérations. La figure 5.31 représente l'évolution du coût estimé en utilisant les politiques jointes disponibles à chaque itération de la résolution, moyenne de 50 000 simulations avec des conditions de départ aléatoires. Les améliorations sont principalement situées dans les 15 premières itérations, et les itérations suivantes ne changent que très peu la qualité de ces politiques. Ainsi, après quelques itérations, la résolution pourra être arrêtée à n'importe quel instant et la dernière politique jointe calculée pourra être utilisée, les gains suivants étant vraisemblablement minimes. Le nombre d'itérations nécessaire avant

d'atteindre des améliorations minimes est *a priori* dépendant de la complexité du réseau et de sa décomposition.

5.4.2 Chaînage des résolutions

L'utilisation d'intervalles pour les actions rend complexe et imprécise l'utilisation de la politique. Trouver la valeur optimale à transférer dans l'intervalle d'action choisi est une tâche difficile qui est semblable au problème initial appliqué sur des domaines réduits.

Une solution possible consiste à effectuer une nouvelle résolution sur le modèle en utilisant une discrétisation plus fine des actions. Cette discrétisation plus fine devient possible grâce à la suppression des actions non utilisées par la politique jointe obtenue. Dès lors que moins de la moitié des actions est utilisée par la politique, les intervalles d'action pourront être divisés en de nouveaux intervalles plus petits pour permettre une nouvelle planification. Cette contrainte a été placée de façon à ce que chaque action conservée puisse être rediscrétisée sans augmenter la taille du modèle. Seules les expérimentations ayant pu obtenir une nouvelle discrétisation ont été réalisées. Dans chaque cas, une politique jointe a été calculée sur la nouvelle discrétisation en utilisant deux types de politiques jointes initiales. La première approche est de construire une nouvelle politique initiale quelconque ; ici elle choisit à tout moment l'action déplaçant le moins d'eau, de façon analogue à la première résolution. La seconde méthode est de réutiliser la politique optimale obtenue lors de la première résolution, permettant *a priori* de minimiser la tâche de coordination. Les résultats obtenus lors de la seconde résolution sont compilés dans la table 5.15.

	Solution initiale		Nouvelle politique		Réutilisation de la politique	
agents	out (%)	avg (%)	out (%)	avg (%)	out (%)	avg (%)
6 agents	0,000	17,13	N/A	N/A	N/A	N/A
7 agents	0,001	15,38	N/A	N/A	N/A	N/A
8 agents	0,000	16,45	0,000	14,71	0,006	16,42
9 agents	0,000	17,27	0,001	16,40	0,000	17,08
10 agents	0,021	17,16	3,753	24,33	0,017	17,10
11 agents	0,023	17,83	0,001	16,20	0,021	17,99
12 agents	0,016	17,27	0,001	17,01	0,016	17,14
13 agents	0,009	16,64	0,000	16,49	0,028	16,12
14 agents	0,007	15,57	0,000	15,76	0,001	15,16

TABLE 5.15 – Évolution des résultats après une seconde résolution

Lorsque les agents réutilisent la politique qu'ils ont précédemment calculée, adaptée à leur nouvel ensemble d'actions, ils peuvent obtenir des résultats de qualité légèrement supérieure. Dans quelques cas, le pourcentage de biefs hors de leur rectangle de navigation augmentera,

mais ceci sera toujours lié à une réduction de l'écart global de la distance relative au NNL. Dans ce cas, l'adaptation de la politique jointe précédemment calculée est réalisée en choisissant de façon arbitraire une action équivalente dans le nouvel ensemble. Or, ce choix peut conduire une politique légèrement dégradée, mais correspondant à une solution localement optimale du nouveau modèle. En moyenne la politique jointe offre de meilleurs résultats, cependant les risques de sortir du rectangle de navigation sont augmentés. Puisque les politiques initiales des agents correspondent à une politique jointe localement optimale du modèle initial, l'algorithme converge *a priori* plus rapidement.

L'utilisation d'une politique initiale quelconque, par rapport à la nouvelle sélection d'actions, pour la seconde résolution permet d'obtenir des améliorations significatives des politiques jointes. Néanmoins, cela peut aussi résulter en une politique de moins bonne qualité (table 5.15 pour 10 agents). Les actions qui n'ont pas été gardées par la première politique jointe calculée ne permettent pas une gestion efficace du réseau, mais peuvent permettre aux agents de se coordonner en améliorant itérativement par compromis leur politique respective.

Ainsi, le chaînage des résolutions permet de réduire la taille des intervalles utilisés pour les actions et conduit globalement à de bons résultats, mais n'offre pas de garantie quant au maintien ou à l'amélioration de la qualité de la gestion.

5.4.3 Analyse des résultats

Les politiques obtenues ont été évaluées uniquement sur des scénarios de gestion correspondant au modèle utilisé par les agents, à l'exception de deux politiques utilisées pour montrer l'impact de l'anticipation des événements pluvieux. Dans l'ensemble, les résultats obtenus montrent une gestion cohérente et acceptable sur les scénarios connus ainsi qu'une capacité à anticiper des perturbations ou des variations incertaines. Cependant, ces résultats dépendent de plusieurs critères :

1. la décomposition en agent ;
2. la discrétisation en intervalles ;
3. la politique initiale.

La décomposition en agents possède une influence sur la qualité de la politique jointe ainsi obtenue. Les tables 5.1 (page 92) et 5.15 ont permis de mettre en avant que le nombre d'agents et donc la décomposition en agents a une influence sur la qualité des résultats. Ainsi, il n'est pas garanti d'obtenir une solution de qualité supérieure en réduisant le nombre d'agents, lorsque la coordination entre agents est principalement locale. Lorsque le nombre d'agents diminue, chaque agent possède potentiellement une connaissance accrue du réseau. Cette connaissance plus importante du réseau est contrebalancée par une réduction proportionnelle des informations communes entre un agent et ses voisins. Intuitivement, le fait pour un agent d'être en difficulté sur un composant qu'il gère seul ne transparaît pas forcément sur les composants en périphérie qu'il partage avec d'autres agents. De ce fait, les politiques adaptées par un agent à ses voisins devront

s'abstraire d'un plus grand nombre d'états ce qui réduira donc la qualité de ces adaptations. La qualité des résultats sera donc dépendante à la fois du nombre d'agents et de la répartition des contrôleurs sur ceux-ci.

La discrétisation en intervalles des états et des actions permet de trouver un équilibre entre la capacité à passer à l'échelle et la précision de la modélisation. Les politiques produites permettent d'obtenir de bons résultats de gestion en choisissant des valeurs aléatoirement dans les intervalles choisis. Néanmoins, un algorithme de recherche des valeurs efficaces pour le pas de temps courant pourrait permettre une amélioration des résultats et une facilitation d'application des politiques ainsi produites.

L'utilisation de politiques initiales proches d'une solution correcte permet d'obtenir un résultat de qualité relative, au risque de tomber trop rapidement dans un optimum local sans amélioration de la politique jointe. L'obtention d'une politique correcte peut cependant être une tâche aussi complexe que le problème originel. Partir d'une politique de mauvaise qualité permet quand même, grâce aux améliorations successives, de trouver des bonnes politiques jointes sans nécessiter de connaissance préalable sur le fonctionnement du réseau. L'intérêt des politiques initiales de mauvaises qualités est de fournir une politique où un grand nombre de composants ou biefs peuvent être améliorés afin d'éviter une convergence trop rapide. Utiliser des politiques initiales aléatoires fait courir le risque de commencer trop proche ou dans un optimum local médiocre.

5.4.4 Industrialisation de l'approche

L'approche proposée pourrait être appliquée pour la gestion de l'eau sur les réseaux de voies navigables. Son application nécessite la présence de capteurs de niveau répartis sur l'ensemble des biefs afin de pouvoir déterminer à chaque instant l'état du réseau. Des mécanismes de communications sont donc nécessaires pour transmettre les informations de niveau aux agents concernés. Il s'agit d'un outil d'aide à la décision dont les performances devraient être confrontées à la gestion réelle et à l'expertise des gestionnaires.

De par sa nature distribuée, le problème possède déjà une décomposition en agents. Un éclusier s'occupe d'un ou plusieurs points de transfert affectant un nombre restreint de biefs. Ainsi définir un agent comme étant un éclusier permet d'obtenir une décomposition du réseau et la capacité d'appliquer les choix de la politique. La discrétisation des actions en intervalles n'est pas une source de problème pour des éclusiers. La politique correspond à des consignes de gestion bornant les quantités d'eau qu'un éclusier pourra déplacer sur un pas temps assez long (de plusieurs minutes à plusieurs heures) en considérant l'état courant du réseau.

Le choix du pas de temps et de l'horizon de prédiction sont dépendants du problème considéré et de la capacité à prendre en compte les événements à venir. Si le but est d'optimiser les conditions de navigation journalières, un pas de temps de plusieurs heures permettant de considérer comme instantané le temps de transfert d'eau et de fournir une bonne estimation du trafic et

de la météo. De même pour l'horizon de prédiction qui se compterait en jours. À l'inverse, si le but est de prévenir des événements importants sur une grande durée telles que des périodes de sécheresses, il pourrait être préférable de prendre des horizons beaucoup plus grand, de l'ordre de plusieurs jours voire du mois. Dans ce cas, une planification en jours pourrait suffire.

Lors de l'utilisation des politiques de gestion produites par l'algorithme OCLP, il ne serait pas forcément nécessaire que celles-ci correspondent au modèle actuel. Il ressort des expérimentations de ce chapitre que les politiques jointes permettent de maintenir des conditions de navigation malgré des conditions d'utilisations différentes de celles de calcul. Par exemple, dans la section 5.2.1 (page 97), une variation inattendue de la demande de navigation de 10% sur l'ensemble de l'horizon de prédiction peut être gérée avec une même politique. De même, des événements pluvieux ponctuels et non critiques, c'est-à-dire qui ne font pas forcément sortir du rectangle de navigation sans anticipation, peuvent être traités grâce à la résilience de la gestion (section 5.2.4 page 109). Ainsi, une politique jointe généraliste pourrait être calculée pour les principales périodes climatiques (normal, pluvieux, sec) avec des calculs ponctuels lors d'événements extrêmes nécessitant une gestion plus précise.

5.5 Conclusion

L'approche proposée dans cette thèse permet de modéliser et de planifier la gestion d'une ressource partagée par un réseau. Elle a été appliquée à la fois sur des réseaux factices et réels de voies navigables dans le but d'évaluer sa capacité à gérer et à anticiper les événements probables et à récupérer de situations très défavorables.

La première expérimentation sur un réseau factice a permis de montrer la capacité à fournir une solution acceptable, mais aussi l'impact de la décomposition en agents sur la qualité de la solution, notamment le fait qu'un faible nombre d'agents n'est pas forcément préférable. Dans un second temps, les expérimentations sur un réseau réel de 3 biefs ont mis en avant la faculté de s'adapter à une application réelle ainsi que la résilience, l'anticipation et la récupération, d'événements probables et inattendus, en prenant en compte diverses incertitudes sur le réseau. Finalement, une expérimentation sur un réseau élargi permet de confirmer la capacité de passage à l'échelle en élargissant la modélisation à un réseau plus grand tout en maintenant la qualité des résultats.

Les principales limitations de l'approche ont été discutées et des solutions possibles ont été proposées. Il est possible d'obtenir une politique jointe de qualité proche de celle finale avant la fin du calcul afin de contourner les problèmes possibles de temps de calcul liés à la construction des fonctions de transition. De même, un chaînage de résolutions est une méthode potentielle pour réduire la taille des intervalles d'actions dans le but de faciliter l'application des politiques jointes par les gestionnaires du réseau.

Conclusion et perspectives

Les travaux de cette thèse visent à produire des politiques pour optimiser la gestion de la ressource en eau dans les réseaux de voies navigables. Afin de rendre cette gestion résiliente, notamment dans un contexte de changement climatique et d'accroissement de la navigation, les événements incertains doivent être pris en compte. Pour cela, une approche générique a été proposée. Elle permet d'obtenir des politiques de gestion d'une ressource partagée, telle que l'eau sur les voies navigables, sur des réseaux complexes de grandes dimensions. Les politiques ainsi produites prennent en compte un maximum d'incertitudes affectant les réseaux considérés afin de maximiser leur résilience.

Une classe de problèmes de gestion prédictive d'une ressource partagée sur un réseau a été définie. Elle représente des systèmes constitués de plusieurs composants répartis sur un réseau. Chaque composant a pour objectif d'optimiser son stockage d'une ressource partagée avec l'ensemble du réseau. Un composant possède des contraintes fortes quant à la quantité de ressource qu'il peut et doit stocker. Des effecteurs sont répartis sur le réseau pour permettre le déplacement entre les composants et ainsi permettre le bon fonctionnement du réseau. Dans ces problèmes, les composants et les effecteurs peuvent être affectés par de nombreuses incertitudes. Les problèmes considérés pouvant être définis sur des domaines continus, une discrétisation de la modélisation est requise. Cette classe de problèmes peut correspondre à diverses applications réelles telles que la gestion de flottes de véhicules, de stockage de l'électricité ou encore des systèmes hydrauliques. Pour représenter ces problèmes, une modélisation à l'aide de processus décisionnels markoviens est utilisée.

Les processus décisionnels markoviens (MDPs) permettent de modéliser la prise de décision dans des systèmes dynamiques et stochastiques. Divers algorithmes ont été définis pour calculer une politique optimale d'un MDP donné. Cependant, cette approche ne permet pas de représenter des problèmes réels à cause d'une explosion combinatoire résultant en une incapacité à passer à l'échelle. Des extensions des MDPs permettant de traiter des problèmes plus imposants ont été proposées. Pour cela, des caractéristiques spécifiques des modèles sont exploitées. Les MDPs factorisés utilisent le fait que certains problèmes sont définis par un ensemble de variables afin de représenter le modèle de façon compacte. Cependant, pour la classe de problèmes considérée, l'évolution des variables (les composants) ne serait pas pleinement indépendante et donc ne peut pas être correctement représentée par des MDPs factorisés. Similairement, les MDPs décompo-

sés exploitent la séparabilité du modèle en plusieurs sous-problèmes faiblement indépendants. Or pour la problématique concernée, la condition de séparabilité de l'espace d'états n'est pas présente. Les MDPs décentralisés avec de faibles interactions (Dec-SIMDP) et les MDPs partiellement observables distribués en réseau (ND-POMDP) sont deux approches multi-agents pour faciliter la résolution de MDP à observations incomplètes. Les ND-POMDPs requièrent l'indépendance d'évolution du modèle de chaque agent avec des objectifs partagés et les Dec-SIMDPs nécessitent un faible nombre d'états en interaction pour les différents agents. Or selon la définition des problèmes traités, tout état du système est une zone d'interaction de plusieurs composants, empêchant ainsi de définir des ensembles d'états indépendants. De ce fait, les ND-POMDPs et les Dec-SIMDPs ne sont pas adaptés à la problématique définie. Une dernière catégorie de solution pour permettre le passage à l'échelle est de ne pas construire le modèle à optimiser, mais d'utiliser un simulateur permettant d'estimer les meilleurs choix possibles. Cependant, l'exploration des solutions peut s'avérer incomplète. Ainsi des événements rares ne sont pas forcément pris en compte nuisant à la qualité de la solution.

Ces méthodes n'étant pas adaptées à la problématique considérée, nous avons proposé une nouvelle approche. Celle-ci permet la distribution de la modélisation et de l'optimisation d'un MDP. Pour cela, les capacités de contrôle du réseau sont réparties entre différents agents. Chaque agent modélise uniquement le sous-réseau qu'il affecte grâce à ses effecteurs. Ce sous-réseau est partagé avec d'autres agents, puisqu'une majorité des composants sont affectés par plusieurs effecteurs. Une coordination entre les agents pour optimiser la gestion de la ressource est donc nécessaire. L'algorithme OCLP (Offline Coordination of Local Plannings) permet aux agents de se coordonner de proche en proche dans le but d'obtenir une solution localement optimale. À chaque itération, chaque agent échange sa politique locale actuelle avec les autres agents concernés. Ceci permet de calculer une nouvelle politique locale prenant en compte les actions des autres agents. Afin d'éviter les changements conflictuels, un principe de coévolution est appliqué ; deux agents affectant un même composant ne peuvent pas changer leur politique locale en même temps. Cet algorithme peut être utilisé jusqu'à ce que les agents convergent vers une politique jointe localement optimale et le détectent. Il est aussi possible d'arrêter à tout instant la résolution et d'utiliser les dernières politiques locales des agents calculées. Pour fonctionner, cet algorithme nécessite un partitionnement du modèle en agent. Un algorithme heuristique simple a été proposé pour déterminer des partitionnements intéressants. L'approche proposée permet un passage à l'échelle de la modélisation et de l'optimisation de la classe de problèmes considérée. La coordination de proche en proche couplée à une prise en compte des incertitudes permet l'obtention d'une politique jointe localement résiliente. Cependant, ni l'optimalité de la solution ni la convergence vers une solution ne peuvent être garanties avec la version actuelle de l'algorithme.

La gestion de la ressource en eau sur les voies navigables est un problème très étudié. Cependant, les approches existantes visent à fournir un contrôle fin de l'eau sur le réseau, en supposant une connaissance parfaite de son évolution au cours du temps. Or, ces réseaux étant très affectés

par les incertitudes (météo, irrigation, trafic, ...), une planification résiliente de ces systèmes nécessite de prendre en compte ces événements incertains. L'algorithme OCLP a ainsi été appliqué pour traiter cette problématique. Dans un premier temps, un réseau factice a été considéré afin d'observer l'impact de la décomposition du réseau en agents sur la solution. Les réseaux de voies navigables sont des systèmes qui sont principalement optimisés localement. De ce fait, la décomposition en agents a un faible impact sur la qualité des résultats, mais a un effet significatif sur les ressources de calculs nécessaires pour obtenir une solution. Dans un second temps, un réseau réel de trois biefs est étudié. Ce réseau est optimisé lors de différentes situations de fonctionnement ; par exemple en prenant en compte des effets possibles du changement climatique et de l'augmentation future du trafic. Dans chaque cas, les politiques jointes calculées fournissent des résultats valides. Lorsque nécessaire, les événements prévisibles sont anticipés et des mécanismes de récupérations permettent de maintenir les conditions de navigation en cas d'occurrence d'un événement imprévu. Une extension de ce réseau à sept biefs est réalisée afin de mettre en avant la capacité à traiter des réseaux plus importants tout en maintenant la qualité des résultats.

Perspectives

À l'issue de cette thèse, plusieurs voies s'ouvrent à nous pour continuer ces travaux de recherche. Certaines pistes pour améliorer la méthodologie semblent prometteuses. De même, une application de l'approche proposée sur d'autres cas d'études pourrait être intéressante.

Méthodologie

La rédaction de cette thèse a permis de mettre en avant plusieurs points sur lesquels notre approche pourrait être améliorée. Actuellement, aucune garantie n'est disponible quant à la convergence de l'algorithme. Dans une des expérimentations, la résolution distribuée a atteint un cycle, l'empêchant ainsi d'atteindre un optimum local. Il pourrait alors être intéressant de déterminer s'il existe des hypothèses particulières sous lesquelles la convergence serait garantie ou à l'inverse impossible. Lorsque la convergence n'est pas possible, des cycles peuvent être atteints. Cependant leur détection peut s'avérer particulièrement lente. Ainsi, il serait intéressant de définir une méthodologie pour détecter plus rapidement des cycles potentiels ou pour les prévenir. Une solution possible pour prévenir l'apparition de cycles serait l'ajout d'aléas lors du calcul du gain des agents pour déterminer lesquels pourront garder leur politique. À chaque itération, chaque agent aurait une faible probabilité de fortement dégrader son gain ; c'est-à-dire l'évaluation de l'amélioration de la solution procurée par la politique qu'il vient de calculer. Par exemple, lors d'une itération le gain d'un agent serait de 1, en supposant qu'il existe une probabilité de 10% que ce gain devienne 0,01. Avec un gain de 1 l'agent aurait la plus grande amélioration de son voisinage et garderait sa politique, alors qu'avec la réduction, un de ses voisins pourrait l'empêcher de garder cette politique. De cette façon, des branchements de l'exploration des solutions deviennent possibles. Ainsi, si les agents atteignent une configuration, qui dans un cas

déterministe serait un cycle, il existera une probabilité qu'il soit brisé après quelques itérations par un agent ne gardant pas sa politique à cause d'une réduction. L'utilisation d'une réduction du gain au lieu de le rendre nul permettrait d'éviter des configurations où aucun agent ne change sa politique locale alors que des améliorations leur sont disponibles. Cependant, cette approche ne permettrait pas de résoudre des cycles où à chaque itération seul un agent possède une amélioration.

L'algorithme utilise les MDPs pour la modélisation des problèmes locaux. Cependant, d'autres outils de modélisation pourraient être appliqués à la classe de problèmes considérée, tant qu'il est possible d'envoyer une estimation de la politique de gestion locale aux autres agents. Par exemple, si les capteurs du réseau ne sont pas performants, l'état du système ne pourrait être que partiellement observable et nécessiterait une modélisation à l'aide de POMDPs. L'algorithme nécessiterait alors des modifications quant à l'échange de politiques entre les agents, mais devrait pouvoir garder la même structure. En continuant sur l'idée que diverses modélisations peuvent être choisies, des agents hétérogènes pourraient être un concept intéressant. Ainsi des MDPs seraient utilisés pour représenter les zones à observations complètes et des POMDPs pour les zones à observations partielles. Un langage commun devrait néanmoins être défini de façon à ce que deux agents différents puissent interpréter les politiques qu'ils reçoivent. Similairement, il pourrait être intéressant de combiner planification classique, à l'aide d'outils de description tels que STRIPS [Fikes and Nilsson, 1971] ou PDDL [McDermott et al., 1998], et probabiliste, ou encore des MDPs à domaine continu avec des MDPs à domaine discret. Il serait ainsi possible de traiter efficacement certaines parties du réseau selon leurs caractéristiques particulières.

Une augmentation des informations échangées entre les agents pourrait être une piste intéressante. Une première idée serait qu'à chaque itération les agents puissent échanger des préférences sur la gestion des composants qu'ils partagent. Par exemple, lors d'une itération si la politique locale d'un agent implique une gestion inefficace d'un composant, l'agent en question enverrait un message aux voisins concernés afin de favoriser ce composant et ainsi y remédier rapidement au problème. Cela se traduirait par la modification des poids sur la fonction de coût, lors de la prochaine optimisation. D'autres connaissances pourraient être partagées dans les voisinages telles que des estimations des fonctions de valeurs des agents, ou les politiques locales qui n'ont pas été communiquées lors d'une itération et qui pourraient contenir des informations sur l'évolution souhaitée des agents.

D'un point de vue plus algorithmique, diverses optimisations sont envisageables afin d'augmenter les performances. Dans l'implémentation actuelle de l'algorithme, à chaque itération, chaque agent recalcule entièrement sa fonction de transition. Il pourrait être intéressant de stocker les résultats intermédiaires de cette construction ne nécessitant que la connaissance interne de l'agent. De cette façon, il ne resterait plus qu'à prendre en compte les effets des estimations des politiques jointes des autres agents. Ceci devrait permettre une réduction importante du temps de calcul, mais en contrepartie nécessiterait un plus grand espace mémoire.

Application

L'approche actuelle produit une politique jointe qui, due à la discrétisation en intervalles des niveaux et des débits, correspond à une planification haut niveau sur une période donnée. De ce fait, son application réelle reste imprécise. Une continuation possible serait de créer une résolution hiérarchique, de façon similaire à ce qui a été proposé dans [Zafra-Cabeza et al., 2011]. Dans un premier temps, l'algorithme OCLP trouve une solution de gestion prenant en compte les différentes incertitudes pour permettre la résilience de l'ensemble du réseau considéré. Cette solution utilise des pas de temps larges pour considérer les déplacements d'eau comme uniformes et instantanés. Elle serait ensuite reprise successivement par un algorithme d'optimisation quadratique et de commande prédictive pour déterminer à chaque instant les valeurs à transférer par chaque effecteur. Les différentes optimisations réduisent successivement la durée des pas de temps et les ensembles de valeurs permises par la solution trouvée par l'algorithme OCLP. Ainsi, il serait possible de proposer une gestion efficace sur des pas de temps faibles tout en étant résiliente sur la durée.

Dans le cadre des voies navigables, les pas de temps considérés sont limités par le temps de déplacement de l'eau sur un bief. Une solution pour réduire ce temps de déplacement serait de diviser chaque bief en *biefs virtuels*. Un tel concept a été proposé dans [Wolfs et al., 2015] afin de réduire la complexité de différentes problématiques de gestion des rivières. Un bief pourrait ainsi être représenté en deux parties, une partie amont et une partie aval, avec des temps de déplacement d'eau diminués de moitié. Cependant, cette solution nécessite de représenter les transferts d'eau entre les deux parties du bief, ce qui requiert une modélisation plus complexe. Évidemment, cette solution pourrait aussi être utilisée pour traiter des systèmes hydrauliques qui ne sont pas divisés en biefs ou en réservoirs.

D'autres applications pourraient être envisagées. Nous avons précédemment discuté des problématiques de gestion de flottes de véhicules ou de stockage d'électricité sur un réseau, mais le problème de platooning ou de convoi routier serait aussi applicable. L'objectif dans le platooning est de minimiser les distances entre les véhicules en coordonnant les changements de vitesse de l'ensemble du convoi. Dans ce cadre, la définition des composants et de la ressource deviendrait plus abstrait. Un composant et sa ressource seraient définis par l'espace entre deux véhicules. Les actions modéliseraient les accélérations et les ralentissements des différents véhicules du convoi.

Bibliographie

[Bellifemine et al., 1999] Bellifemine, F., Poggi, A., and Rimassa, G. (1999). JADE–a FIPA-compliant agent framework. In *Proceedings of PAAM*, volume 99, page 33. London.

[Bellman, 1957] Bellman, R. (1957). A markovian decision process. *Journal of Mathematics and Mechanics*, 6(4) :679–684.

[Bernstein et al., 2002] Bernstein, D. S., Givan, R., Immerman, N., and Zilberstein, S. (2002). The complexity of decentralized control of Markov decision processes. *Mathematics of operations research*, 27(4) :819–840.

[Beuthe et al., 2012] Beuthe, M., Jourquin, B., Urbain, N., Bruinsma, F., Lingemann, I., Ubbels, B., and Heumen, E. V. (2012). Estimating the impacts of water depth and new infrastructures on transport by inland navigation : A multimodal approach for the Rhine corridor. *Procedia - Social and Behavioral Sciences - Proceedings of EWGT2012 - 15th Meeting of the EURO Working Group on Transportation*, 54 :387 – 401.

[Beuthe et al., 2014] Beuthe, M., Jourquin, B., Urbain, N., Lingemann, I., and Ubbels, B. (2014). Climate change impacts on transport on the Rhine and Danube : A multimodal approach. *Transportation Research Part D : Transport and Environment*, 27 :6 – 11.

[Beynier, 2006] Beynier, A. (2006). *Une contribution à la résolution des processus décisionnels de Markov décentralisés avec contraintes temporelles.* PhD thesis, Université de Caen.

[Bichot and Siarry, 2011] Bichot, C.-E. and Siarry, P. (2011). *Graph partitioning.* Wiley-ISTE.

[Boé et al., 2009] Boé, J., Terray, L., Martin, E., and Habets, F. (2009). Projected changes in components of the hydrological cycle in French river basins during the 21st century. *Water Resources Research*, 45(8).

[Boutilier, 1996] Boutilier, C. (1996). Planning, learning and coordination in multiagent decision processes. In *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, pages 195–210. Morgan Kaufmann Publishers Inc.

[Boutilier et al., 1997] Boutilier, C., Brafman, R. I., and Geib, C. (1997). Prioritized goal decomposition of Markov decision processes : Toward a synthesis of classical and decision theoretic planning. In *IJCAI*, pages 1156–1162.

[Boutilier et al., 1999] Boutilier, C., Dean, T., and Hanks, S. (1999). Decision-theoretic planning : Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11(1) :94.

[Boutilier et al., 1995] Boutilier, C., Dearden, R., and Goldszmidt, M. (1995). Exploiting structure in policy construction. In *IJCAI*, volume 14, pages 1104–1113.

[Boutilier et al., 2000] Boutilier, C., Dearden, R., and Goldszmidt, M. (2000). Stochastic dynamic programming with factored representations. *Artificial intelligence*, 121(1) :49–107.

[Brand et al., 2012] Brand, C., Tran, M., and Anable, J. (2012). The UK transport carbon model : An integrated life cycle approach to explore low carbon futures. *Energy Policy*, 41 :107–124.

[Camacho and Alba, 2013] Camacho, E. F. and Alba, C. B. (2013). *Model predictive control*. Springer Science & Business Media.

[Cassandra et al., 1997] Cassandra, A., Littman, M. L., and Zhang, N. L. (1997). Incremental pruning : A simple, fast, exact method for partially observable Markov decision processes. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 54–61. Morgan Kaufmann Publishers Inc.

[Chades et al., 2002] Chades, I., Scherrer, B., and Charpillet, F. (2002). A heuristic approach for solving decentralized-POMDP : Assessment on the pursuit problem. In *SAC '02 : Proceedings of the 2002 ACM symposium on Applied computing*, pages 57–62, Madrid, Spain. ACM.

[Chauveau et al., 2013] Chauveau, M., Chazot, S., Perrin, C., Bourgin, P.-Y., Sauquet, E., Vidal, J.-P., Rouchy, N., Martin, E., David, J., Norotte, T., Maugis, P., and De Lacaze, X. (2013). Quels impacts des changements climatiques sur les eaux de surface en France à l'horizon 2070 ? *La Houille Blanche*, (4) :5–15.

[Dean and Lin, 1995] Dean, T. and Lin, S.-H. (1995). Decomposition techniques for planning in stochastic domains. In *IJCAI*, volume 2, page 3.

[Desquesnes et al., 2018a] Desquesnes, G., Alves, D., Lozenguez, G., Doniec, A., and Duviella, E. (2018a). Simulation architecture based on distributed MDP for inland waterway management. In *13th International Conference on Hydroinformatics, 2018*.

[Desquesnes et al., 2017a] Desquesnes, G., Lozenguez, G., Arnaud, D., and Duviella, E. (2017a). Vers une distribution des MDP à grande échelle : étude de cas des voies navigables. *Revue des Sciences et Technologies de l'Information-Série RIA : Revue d'Intelligence Artificielle*, 31(1-2) :183–205.

[Desquesnes et al., 2016a] Desquesnes, G., Lozenguez, G., Doniec, A., and Duviella, E. (2016a). Dealing with large MDPs, case study of waterway networks supervision. In *Advances in Practical Applications of Scalable Multi-agent Systems. The PAAMS Collection*, pages 48–59. Springer.

[Desquesnes et al., 2016b] Desquesnes, G., Lozenguez, G., Doniec, A., and Duviella, É. (2016b). MDP à grande échelle : étude de cas des voies navigables. In *Journées Applications Pratiques en Intelligence Artificielle (APIA) pendant le vingtième congrès national sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'16)*.

[Desquesnes et al., 2016c] Desquesnes, G., Lozenguez, G., Doniec, A., and Duviella, É. (2016c). Planning large systems with MDPs : case study of inland waterways supervision. *ADCAIJ : Advances in Distributed Computing and Artificial Intelligence Journal*, 5(4) :71–84.

[Desquesnes et al., 2017b] Desquesnes, G., Lozenguez, G., Doniec, A., and Duviella, E. (2017b). Coordination distribuée et hors-ligne de planifications locales. In *Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite de systèmes (JFPDA 2017)*.

[Desquesnes et al., 2017c] Desquesnes, G., Lozenguez, G., Doniec, A., and Duviella, É. (2017c). Distributed MDP for water resources planning and management in inland waterways. *IFAC-PapersOnLine*, 50(1) :6576–6581.

[Desquesnes et al., 2018b] Desquesnes, G., Lozenguez, G., Doniec, A., and Duviella, É. (2018b). Large Markov decision processes based management strategy of inland waterways in uncertain context. In *Advances in Hydroinformatics*, pages 3–20. Springer.

[Desquesnes et al., 2016d] Desquesnes, G., Nouasse, H., Lozenguez, G., Doniec, A., and Duviella, E. (2016d). A global approach for investigating resilience in inland navigation network dealing with climate change context. *Procedia Engineering*, 154 :718–725.

[Dibangoye et al., 2014] Dibangoye, J. S., Amato, C., Buffet, O., and Charpillet, F. (2014). Exploiting separability in multiagent planning with continuous-state MDPs. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1281–1288. International Foundation for Autonomous Agents and Multiagent Systems.

[Ducharne et al., 2010] Ducharne, A., Habets, F., Pagé, C., Sauquet, E., Viennot, P., Déqué, M., Gascoin, S., Hachour, A., Martin, E., Oudin, L., Terray, L., and Thiéry, D. (2010). Climate change impacts on water resources and hydrological extremes in northern France. In *XVIII Conference on Computational Methods in Water Resources*.

[Duviella, 2005] Duviella, E. (2005). Conduite réactive des systèmes dynamiques étendus à retards variables. In *Cas des réseaux hydrographiques*. ENI Tarbes France.

[Duviella et al., 2018] Duviella, E., Doniec, A., and Nouasse, H. (2018). Adaptive water-resource allocation planning of inland waterways in the context of global change. *Journal of Water Resources Planning and Management*, 144(9).

[Duviella et al., 2016] Duviella, E., Nouasse, H., Doniec, A., and Chuquet, K. (2016). Dynamic optimization approaches for resource allocation planning in inland navigation networks. *ETFA'2016, Berlin, Allemagne, September 6-9*.

[Ehrlich and Raven, 1964] Ehrlich, P. R. and Raven, P. H. (1964). Butterflies and plants : a study in coevolution. *Evolution*, 18(4) :586–608.

[Ficchì et al., 2015] Ficchì, A., Raso, L., Dorchies, D., Pianosi, F., Malaterre, P.-O., Van Overloop, P.-J., and Jay-Allemand, M. (2015). Optimal operation of the multireservoir system in the Seine river basin using deterministic and ensemble forecasts. *Journal of Water Resources Planning and Management*, 142(1) :05015005.

[Fikes and Nilsson, 1971] Fikes, R. E. and Nilsson, N. J. (1971). STRIPS : A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4) :189–208.

[Franklin and Graesser, 1996] Franklin, S. and Graesser, A. (1996). Is it an agent, or just a program ? : A taxonomy for autonomous agents. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 21–35. Springer.

[Guestrin et al., 2003] Guestrin, C., Koller, D., Parr, R., and Venkataraman, S. (2003). Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, 19 :399–468.

[Guestrin and Koller, 2003] Guestrin, C. E. and Koller, D. (2003). *Planning under uncertainty in complex structured environments*. PhD thesis, Stanford University.

[Hoey et al., 1999] Hoey, J., St-Aubin, R., Hu, A., and Boutilier, C. (1999). SPUDD : Stochastic planning using decision diagrams. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 279–288. Morgan Kaufmann Publishers Inc.

[Howard, 1964] Howard, R. A. (1964). *Dynamic programming and Markov processes*. Wiley for The Massachusetts Institute of Technology.

[Kaelbling et al., 1998] Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2) :99–134.

[Kearns et al., 2002] Kearns, M., Mansour, Y., and Ng, A. Y. (2002). A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine learning*, 49(2-3) :193–208.

[Kocsis and Szepesvári, 2006] Kocsis, L. and Szepesvári, C. (2006). Bandit based Monte-Carlo planning. In *European conference on machine learning*, pages 282–293. Springer.

[Kocsis et al., 2006] Kocsis, L., Szepesvári, C., and Willemson, J. (2006). Improved Monte-Carlo search. *Univ. Tartu, Estonia, Tech. Rep*, 1.

[Lozenguez et al., 2012] Lozenguez, G., Adouane, L., Beynier, A., Mouaddib, A.-I., and Martinet, P. (2012). Map partitioning to approximate an exploration strategy in mobile robotics. *Multiagent and Grid Systems*, 8(3) :275–288.

[Madani et al., 1999] Madani, O., Hanks, S., and Condon, A. (1999). On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *AAAI/IAAI*, pages 541–548.

[Maestre and Negenborn,] Maestre, J. M. and Negenborn, R. R. *Distributed model predictive control made easy*, volume 69. Springer.

[Mallidis et al., 2012] Mallidis, I., Dekker, R., and Vlachos, D. (2012). The impact of greening on supply chain design and cost : a case for a developing region. *Journal of Transport Geography*, 22 :118–128.

[Mansley et al., 2011] Mansley, C. R., Weinstein, A., and Littman, M. L. (2011). Sample-based planning for continuous action Markov decision processes. In *ICAPS*.

[Marecki et al., 2008] Marecki, J., Gupta, T., Varakantham, P., Tambe, M., and Yokoo, M. (2008). Not all agents are equal : Scaling up distributed POMDPs for agent networks. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '08, pages 485–492. International Foundation for Autonomous Agents and Multiagent Systems.

[Marecki et al., 2006] Marecki, J., Topol, Z., and Tambe, M. (2006). A fast analytical algorithm for MDPs with continuous state spaces. In *AAMAS-06 Proceedings of 8th Workshop on Game Theoretic and Decision Theoretic Agents*.

[McDermott et al., 1998] McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D., and Wilkins, D. (1998). PDDL-the planning domain definition language.

[Melo and Veloso, 2013] Melo, F. S. and Veloso, M. (2013). *Heuristic Planning for Decentralized MDPs with Sparse Interactions*, pages 329–343. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Meuleau et al., 1998] Meuleau, N., Hauskrecht, M., Kim, K.-E., Peshkin, L., Kaelbling, L. P., Dean, T. L., and Boutilier, C. (1998). Solving very large weakly coupled Markov decision processes. In *AAAI/IAAI*, pages 165–172.

[Mihic et al., 2011] Mihic, S., Golusin, M., and Mihajlovic, M. (2011). Policy and promotion of sustainable inland waterway transport in Europe – Danube river. *Renewable and Sustainable Energy Reviews*, 15(4) :1801–1809.

[Morris, 1993] Morris, P. (1993). The breakout method for escaping from local minima. In *AAAI*, volume 93, pages 40–45.

[Nair et al., 2005] Nair, R., Varakantham, P., Tambe, M., and Yokoo, M. (2005). Networked distributed POMDPs : A synthesis of distributed constraint optimization and POMDPs. In *National Conference on Artificial Intelligence*, page 7.

[Nasir et al., 2016] Nasir, H. A., Garatti, S., and Weyer, E. (2016). Scenario based stochastic MPC schemes for rivers with feasibility ssurance. In *2016 European Control Conference (ECC)*, pages 1928–1933.

[Nasir et al., 2018] Nasir, H. A., Zhao, T., Carè, A., Wang, Q. J., and Weyer, E. (2018). Efficient river management using stochastic MPC and ensemble forecast of uncertain in-flows. *IFAC-PapersOnLine*, 51(5) :37–42.

[Nouasse et al., 2016] Nouasse, H., Doniec, A., Lozenguez, G., Duviella, E., Chiron, P., Archimède, B., and Chuquet, K. (2016). Constraint satisfaction problem based on flow graph to study the resilience of inland navigation networks in a climate change context. *IFAC-PapersOnLine*, 49(12) :331–336.

[Nouasse et al., 2015] Nouasse, H., Rajaoarisoa, L., Doniec, A., Duviella, E., Chuquet, K., Chiron, P., and Archimède, B. (2015). Study of drought impact on inland navigation systems based on a flow network model. In *Information, Communication and Automation Technologies (ICAT), 2015 XXV International Conference on*, pages 1–6. IEEE.

[Papadimitriou and Tsitsiklis, 1987] Papadimitriou, C. H. and Tsitsiklis, J. N. (1987). The complexity of Markov decision processes. *Mathematics of operations research*, 12(3) :441–450.

[Parr, 1998] Parr, R. (1998). Flexible decomposition algorithms for weakly coupled Markov decision problems. In *14th Conference on Uncertainty in Artificial Intelligence*, page 422–430.

[Prodan et al., 2017] Prodan, I., Lefevre, L., Genon-Catalot, D., and Nguyen, L.-D.-L. (2017). Distributed model predictive control of irrigation systems using cooperative controllers. *IFAC-PapersOnLine*, 50(1) :6564–6569.

[Puterman, 1994] Puterman, M. L. (1994). *Markov decision processes : Discrete stochastic dynamic programming*. John Wiley & Sons, Inc.

[Radoszycki et al., 2015] Radoszycki, J., Peyrard, N., and Sabbadin, R. (2015). Solving F^3 MDPs : Collaborative multiagent Markov decision processes with factored transitions, rewards and stochastic policies. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 3–19. Springer.

[Russell and Norvig, 2003] Russell, S. J. and Norvig, P. (2003). *Artificial intelligence : a modern approach*. Prentice Hall Series.

[Sabbadin, 2002] Sabbadin, R. (2002). Graph partitioning techniques for Markov Decision Processes decomposition. In *15th Eureopean Conference on Artificial Intelligence*, page 670–674.

[Şahin and Morari, 2010] Şahin, A. and Morari, M. (2010). *Decentralized Model Predictive Control for a Cascade of River Power Plants*, pages 463–485. Springer Netherlands, Dordrecht.

[Segovia et al., 2018a] Segovia, P., Desquesnes, G., Doniec, A., Duviella, E., Lozenguez, G., Nejjari, F., Puig, V., and Rajaoarisoa, L. (2018a). Management tools to study and to deal with effects of climate change on inland waterways. In *TRA 2018*.

[Segovia et al., 2018b] Segovia, P., Rajaoarisoa, L., Nejjari, F., Duviella, E., and Puig, V. (2018b). Distributed input-delay model predictive control of inland waterways. In *13th International Conference on Hydroinformatics, 2018*.

[Seuken and Zilberstein, 2012] Seuken, S. and Zilberstein, S. (2012). Improved memory-bounded dynamic programming for decentralized POMDPs. *arXiv preprint arXiv :1206.5295*.

[Sigaud and Buffet, 2008] Sigaud, O. and Buffet, O. (2008). *Processus décisionnels de Markov en intelligence artificielle*. Lavoisier-Hermes Science Publications.

[Singh and Cohn, 1998] Singh, S. P. and Cohn, D. (1998). How to dynamically merge Markov decision processes. In *Advances in neural information processing systems*, pages 1057–1063.

[Smallwood and Sondik, 1973] Smallwood, R. D. and Sondik, E. J. (1973). The optimal control of partially observable Markov processes over a finite horizon. *Operations research*, 21(5) :1071–1088.

[Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning : An introduction*. MIT press.

[Tsang, 1993] Tsang, E. (1993). Chapter 1 - introduction. In Tsang, E., editor, *Foundations of Constraint Satisfaction*, pages 1 – 30. Academic Press.

[Van Overloop et al., 2010] Van Overloop, P., Negenborn, R., De Schutter, B., and Van De Giesen, N. (2010). Predictive control for national water flow optimization in the Netherlands. In *Intelligent infrastructures*, pages 439–461. Springer.

[Walker et al., 2004] Walker, B., Holling, C., Carpenter, S., and Kinzig, A. (2004). Resilience, adaptability and transformability in social-ecological systems. *Ecology and Society*, 9(2).

[Wang et al., 2017] Wang, Y., Cembrano, G., Puig, V., Urrea, M., Romera, J., Saporta, D., Valero, J., and Quevedo, J. (2017). Optimal management of Barcelona water distribution network using non-linear model predictive control. *IFAC-PapersOnLine*, 50(1) :5380–5385.

[Wolfs et al., 2015] Wolfs, V., Meert, P., and Willems, P. (2015). Modular conceptual modelling approach and software for river hydraulic simulations. *Environmental Modelling & Software*, 71 :60–77.

[Wooldridge and Jennings, 1995] Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents : Theory and practice. *The knowledge engineering review*, 10(2) :115–152.

[Yokoo and Hirayama, 1996] Yokoo, M. and Hirayama, K. (1996). Distributed breakout algorithm for solving distributed constraint satisfaction problems. In *Proceedings of the Second International Conference on Multi-Agent Systems*, pages 401–408.

[Zafra-Cabeza et al., 2011] Zafra-Cabeza, A., Maestre, J., Ridao, M. A., Camacho, E. F., and Sánchez, L. (2011). A hierarchical distributed model predictive control approach to irrigation canals : A risk mitigation perspective. *Journal of Process Control*, 21(5) :787–799.

Distribution de Processus Décisionnels Markoviens pour une gestion prédictive d'une ressource partagée : Application aux voies navigables des Hauts-de-France dans le contexte incertain du changement climatique

Les travaux de cette thèse visent à mettre en place une gestion prédictive sous incertitudes de la ressource en eau pour les réseaux de voies navigables. L'objectif est de proposer un plan de gestion de l'eau pour optimiser les conditions de navigation de l'ensemble du réseau supervisé sur un horizon spécifié. La solution attendue doit rendre le réseau résilient aux effets probables du changement climatique et aux évolutions du trafic fluvial.

Dans un premier temps, une modélisation générique d'une ressource distribuée sur un réseau est proposée. Celle-ci, basée sur les processus décisionnels markoviens, prend en compte les nombreuses incertitudes affectant les réseaux considérés. L'objectif de cette modélisation est de couvrir l'ensemble des cas possibles, prévus ou non, afin d'avoir une gestion résiliente de ces réseaux. La seconde contribution consiste en une distribution du modèle sur plusieurs agents afin de permettre son passage à l'échelle. Ceci consiste en une répartition des capacités de contrôle du réseau entre les agents. Chaque agent ne possède ainsi qu'une connaissance locale du réseau supervisé. De ce fait, les agents ont besoin de se cordonner pour proposer une gestion efficace du réseau. Une résolution itérative, avec échanges de plans temporaires de chaque agent, est utilisée pour l'obtention de politiques de gestion locales à chaque agent.

Finalement, des expérimentations ont été réalisées sur des réseaux réels de voies navigables françaises pour observer la qualité des solutions produites. Plusieurs scénarios climatiques différents ont été simulés pour tester la résilience des politiques produites.

Distributing Markovian Decision Processes for a predictive management of a shared resource : Application to the Hauts-de-France waterways in the uncertain context of climate change

The work of this thesis aims to introduce and implement a predictive management under uncertainties of the water resource for inland waterway networks. The objective is to provide a water management plan to optimize the navigation conditions of the entire supervised network over a specified horizon. The expected solution must make the network resilient to probable effects of the climate change and changes in waterway traffic.

Firstly, a generic modeling of a resource distributed on a network is proposed. This modeling, based on Markovian Decision Processes, takes into account the numerous uncertainties affecting considered networks. The objective of this modeling is to cover all possible cases, foreseen or not, in order to have a resilient management of those networks. The second contribution consists in a distribution of the model over several agents to facilitate the scaling. This consists of a repartition of the network's control capacities among the agents. Thus, each agent has only local knowledge of the supervised network. As a result, agents require coordination to provide an efficient management of the network. An iterative resolution, with exchanges of temporary plans from each agent, is used to obtain local management policies for each agent.

Finally, experiments were carried out on realistic and real networks of the French waterways to observe the quality of the solutions produced. Several different climatic scenarios have been simulated to test the resilience of the produced policies.