

**THÈSE DE DOCTORAT DE
L'UNIVERSITÉ PARIS 13**

École doctorale Sciences, Technologies, Santé "Galilée"

Présentée et soutenue publiquement par

Lise-Marie VEILLON

le 8 mars 2018

Pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PARIS 13

Spécialité : **Informatique**

***Apprentissage artificiel collectif;
aspects dynamiques et structurels***

Composition du jury :

M.	Henry SOLDANO	MCF-hdr	Directeur de thèse
M.	Gauvain BOURGNE	MCF	Co-encadrant
Mme.	Lynda TAMINE-LECHANI	PR	Rapporteur
M.	Bruno ZANUTTINI	MCF-hdr	Rapporteur
M.	Frédéric ROUPIN	PR	Examineur
Mme.	Amal EL FALLAH SEGHRUCHNI	PR	Examineur
M.	Grégory BONNET	MCF	Invité

Résumé

L'apprentissage collectif dans un système multi-agents considère comment une communauté constituée d'agents autonomes partageant un même objectif d'apprentissage peut bénéficier, au niveau individuel et collectif, d'échanges structurés d'informations. Les agents, reliés par un réseau de communication, ont tous la faculté de percevoir des observations, appelées exemples d'apprentissage. Cette thèse s'appuie sur un protocole pré-existant, SMILE (Sound-Multi-agent-Incremental-LEarning), qui organise un échange parcimonieux d'exemples et d'hypothèses. Ce protocole garantit, dans le cas d'agents tous connectés, que les agents obtiennent une hypothèse qui tient compte de tous les exemples perçus par l'ensemble des agents. Il existe des variantes séquentielles de ce protocole qui mettent en place une propagation de l'information pour offrir les mêmes garanties dans un réseau, non pas complet mais, connexe.

Cette thèse apporte deux nouveaux éclairages sur l'apprentissage artificiel collectif. Une première étude montre l'influence de la structure du réseau sur l'apprentissage avec un protocole dont les communications sont limitées au voisinage, sans propagation. Une seconde contribution présente et analyse un nouveau protocole, Waves, qui préserve les garanties de SMILE et dont les interactions en parallèle rendent l'apprentissage en réseau plus dynamique. Ce protocole est évalué en détail, dans un contexte simplifié de tour par tour, ce qui permet de le raffiner par la suite avec diverses améliorations. Il est cependant conçu pour s'adapter à un apprentissage en ligne et une acquisition non limitée, ni en temps ni en nombre, de nouveaux exemples par la communauté.

Abstract

Collective learning in multi-agent systems considers how a community of autonomous agents sharing a learning purpose may benefit from exchanging information to learn efficiently as a community as well as individuals. The community forms a communication network where each agent may access observations, called learning examples. This thesis is based on a former protocol, SMILE (Sound-Multi-agent-Incremental-LEarning), which sets up parsimonious examples and hypotheses exchanges between agents. In a fully connected community, this protocol guarantees an agent's hypothesis takes into account all the examples obtained by the community. Some sequential protocols add propagation to SMILE in order to extend this consistency guarantee to other connected networks.

This thesis contribution to the artificial collective learning field is two-fold. First, we investigate the influence of network structures on learning in networks when communication is limited to neighbourhood without further information propagation. Second, we present and analyze a new protocol, Waves, with SMILE's guarantees and a more dynamic learning process thanks to its execution in parallel. The evaluation of this protocol in a simple turn-based setting gives the opportunity to improve it here in multiple ways. It is however meant to be used with online learning without any restriction on the acquisition rate of new examples, neither on speed nor number.

Remerciements

La recherche de la connaissance est une grande aventure, la science, un vaste monde riche et surprenant.

Merci à tous ceux qui m'ont accompagnée, encouragée et soutenue pendant cette thèse.

C'est ce partage, tant humain que scientifique, qui donne cœur aux travaux présentés ici et viens clore ce chapitre d'histoire.

Table des matières

1	Introduction	7
1.1	Introduction	7
1.2	Motivation	7
1.3	Articulation du document	8
2	Contexte et état de l'art	10
2.1	Apprentissage	10
2.1.1	Processus d'apprentissage supervisé	11
2.1.2	Apprentissage réalisable et exact	15
2.1.3	Apprentissage symbolique	16
2.1.4	Algorithme d'apprentissage incrémental ascendant	20
2.1.5	Agents apprenants	22
2.1.6	Apprentissage par requêtes	25
2.2	Apprentissage collaboratif	26
2.2.1	Identifier une forme d'apprentissage multi-agents	26
2.2.2	La diversité des apprentissages collaboratifs	27
2.2.3	Apprentissage distribué et fouille de données	31
2.2.4	Collaborations en recommandations	33
2.2.5	Approches psychologiques et sociales	35
2.2.6	Problématiques économiques en sciences régionales	37
2.2.7	Argumentation	38
2.2.8	Apprentissage collectif	40
2.3	Réseau	43
2.3.1	Les structures de réseaux	43
2.3.2	Influence de la structure de réseau	48
2.3.3	Les dynamiques de réseau	50
3	SMILE	55
3.1	Protocole pour des agents tous connectés	55
3.1.1	Communication selon le protocole SMILE	55
3.1.2	Évaluation d'un apprentissage	57

3.1.3	Résultats du protocole SMILE	58
3.1.4	Variantes	59
3.2	Adaptation séquentielle au réseau	63
3.2.1	Transition vers le réseau	63
3.2.2	Protocoles réseaux séquentiels	64
4	Apprentissage d'un réseau dont les communications sont limitées au voisinage immédiat	69
4.1	Variables de l'étude	69
4.1.1	Exemple détaillé de révision de groupe	70
4.1.2	Variables du protocole SMILE	70
4.2	Résultats d'apprentissage limité au voisinage immédiat	73
4.2.1	Des graphes similaires : Small-World	73
4.2.2	Des graphes de même densité	74
4.2.3	Des graphes de même densité et distance moyenne, les multipôles	77
4.2.4	Évolution de l'influence des paramètres au cours de l'apprentissage	78
4.3	Propagation d'exemples et d'hypothèses	79
4.3.1	Effet théorique	80
4.3.2	Différents profils selon le diamètre	80
4.3.3	Adoption et mémorisation dépendent de différents paramètres	82
5	Apprentissage réseau SMA-consistant : envisager une approche parallèle	86
5.1	Pourquoi apprendre en parallèle	86
5.2	Apparition de multiples hypothèses	87
5.2.1	Utilisation du broadcast en réseau avec le protocole Variable-Root	87
5.2.2	Problème des exemples multiples	87
5.3	Conflits entre hypothèses	88
5.4	Dépasser la limite du séquentiel	88
5.4.1	Choisir immédiatement une des hypothèses	89
5.4.2	Maintenir plusieurs hypothèses	90
6	Waves, un Protocole réseau parallèle	91
6.1	Construction et propriétés	91
6.1.1	Propriétés du protocole Waves	92
6.1.2	Formation du protocole Waves	96
6.1.3	Version algorithmique du protocole Waves	102

6.1.4	Exemples d'états mémoire d'agents Waves lors d'un tour	104
6.2	Résultats expérimentaux	107
6.2.1	Paramètres	108
6.2.2	Résultats et analyse	109
6.2.3	Bilan du protocole Waves initial	118
6.3	Améliorations	119
6.3.1	Fusion d'hypothèses par classe d'équivalence	119
6.3.2	Économie de critiques	124
6.3.3	Oubli	129
7	Compléments et Discussion	135
7.1	Relaxations de Waves	135
7.1.1	L'outil du tour par tour	135
7.1.2	Waves en continu	136
7.1.3	Quelle hypothèse proposer ?	137
7.1.4	Relaxations	138
7.2	Effets d'une confiance non absolue	139
7.2.1	Incohérence de comportement, mais cohérence de protocole	139
7.2.2	Envisager une τ -acceptabilité	140
7.2.3	Approche perso-consistante, SMA-acceptable	141
7.2.4	Approche perso-acceptable, SMA-acceptable	142
7.2.5	Apprentissage réaliste	142
7.3	Prolongements et analyses complémentaires	142
7.3.1	Résultats de Waves	142
7.3.2	Réseaux de structure dynamique	143
7.3.3	Répartition non uniforme des exemples	143
8	Conclusion	145
A	graphes de 50 sommets	160

Chapitre 1

Introduction

1.1 Introduction

Ce document de thèse concerne l'analyse de l'émergence et de l'évolution d'un savoir collectif. Cette évolution, abordée ici dans un contexte d'intelligence artificielle, fait écho à celle de la connaissance au sein de communautés ou de réseaux sociaux. Il s'agit principalement d'un travail de modélisation et d'exploration de l'apprentissage collectif. L'optimisation de ces modèles vient en second plan enrichir leur analyse.

L'apprentissage collectif est une manière d'apprendre individuellement tout en partageant ses connaissances et en mettant à profit celles des autres individus. L'apprentissage multi-agents offre la possibilité d'une dimension collective. Il est utilisé dans cette étude pour mettre en place une collaboration bienveillante entre agents autonomes. Chaque agent peut réaliser un apprentissage seul. Tout autre agent avec qui il peut communiquer peut cependant bénéficier de ses connaissances sans restrictions ni falsifications au moyen d'un échange de messages. Cette étude, qui se base sur un protocole pré-existant, SMILE, propose d'en approfondir l'analyse et d'en repousser les limites. Le but étant autant de comprendre les phénomènes à l'œuvre lors d'un apprentissage collectif que de l'envisager dans de nouvelles conditions ou encore d'avoir une certaine considération d'efficacité.

1.2 Motivation

L'apprentissage en intelligence artificielle (IA) fait de nombreux progrès. On peut citer par exemple, en apprentissage dans l'adversité, l'arrivée d'Al-

phaGO [SHM⁺16] en 2016 qui a appris à jouer et gagner au go contre d'autres IA ainsi que contre les meilleurs humains. Nous laisserons cependant de côté l'apprentissage dans l'adversité pour nous intéresser plus particulièrement à l'apprentissage dans des Systèmes Multi-Agents (SMA) collaboratifs. Là aussi, les progrès sont nombreux et très divers. Plusieurs approches multi-agents envisagent soit une division du travail impliquant la spécialisation de chaque agent, soit une méthode qui ne peut être réalisée qu'à plusieurs. Le premier cas répond à un besoin d'optimisation. Il permet un traitement efficace et rapide d'une tâche d'apprentissage, comme peuvent le faire des réseaux de neurones. Ces derniers peuvent être assimilés à un système multi-agents simple et très spécialisé. Le second cas peut être illustré par la tâche d'apprentissage de déplacer un objet trop lourd pour un seul robot. Le robot doit donc apprendre, en groupe, une tâche qui n'est ensuite réalisable qu'en groupe.

Le sujet de cette thèse se démarque des cas précédents et concerne l'apprentissage collectif. Moins étudiée, cette forme d'apprentissage correspond à d'autres problématiques que l'on rencontre également au quotidien. L'apprentissage collectif est un phénomène courant au sein de groupes de travail. Il permet de représenter, par exemple, une communauté de chercheurs travaillant dans un même domaine. Cet apprentissage est influencé par la présence de liens entre les individus, la forme des connaissances échangées ou encore l'attitude des individus. Conformément à l'objectif de modélisation et d'analyse de cette thèse, nous utilisons une approche multi-agents pour étudier les paramètres qui structurent les échanges entre individus. Nous proposons également un nouveau protocole, plus dynamique, qui modélise l'apprentissage simultané de plusieurs apprenants au sein d'un même réseau.

1.3 Articulation du document

Ce document commence, chapitre 2, par la contextualisation de cette thèse dans un état de l'art sur l'apprentissage multi-agents au sein du domaine de l'intelligence artificielle. L'apprentissage collectif est en particulier situé dans le cadre de l'apprentissage collaboratif. En raison des difficultés causées par la limitation des communications d'une part et la simultanéité de ces communications d'autre part le document s'articule ensuite en ajoutant les difficultés progressivement. Dans le chapitre 3, est exposé le protocole SMILE initial et ses diverses variantes, dont son approche séquentielle des réseaux. Le chapitre 4 montre l'influence des restrictions de communication lors de l'utilisation d'un protocole sans propagation dans un réseau. Le cha-

pitre 5 met en relation les solutions d'apprentissage parallèles sans limitation de communication et séquentielles avec limitation de communication. Il permet de mieux comprendre la difficulté du fonctionnement en parallèle dans un réseau. Le chapitre 6 propose un nouveau protocole, Waves, fonctionnant en parallèle dans un réseau. Ce protocole comprend des garanties théoriques et des résultats expérimentaux ainsi que plusieurs améliorations. Les deux derniers chapitres permettent de lier ces résultats à des travaux futurs et de conclure sur ce travail de thèse.

Chapitre 2

Contexte et état de l'art

2.1 Apprentissage

Avant de se préoccuper de comment apprendre ensemble, nous allons définir l'apprentissage considéré, comment il est réalisé et ce que représente un agent.

Nous nous intéressons à l'apprentissage en tant que processus d'induction supervisée. On peut reprendre la définition proposée par Cornuéjol et Miclet [CM11] dans leur livre "apprentissage artificiel concepts et algorithmes" au chapitre 1 :

“Qu'est-ce que l'induction ? C'est le processus par lequel on tire des lois de portée générale en partant de l'observation de cas particuliers.”

De façon générale, l'apprentissage se base sur des données représentant des cas particuliers, que l'on appellera *exemples*.

Il est possible de trouver les grandes catégories d'apprentissage dans divers livres sur le sujet [Bis06, CM11]. On distingue l'apprentissage supervisé de l'apprentissage non supervisé.

Dans le cas de l'apprentissage supervisé, l'objectif est clairement défini : il faut pour chaque notion à apprendre, déterminer la fonction qui associe la bonne valeur à chaque exemple. L'apprentissage s'effectue à partir d'exemples étiquetés avec la valeur attendue. Si la fonction recherchée a un nombre fini de valeurs de sortie, il s'agit d'un apprentissage en classification. Si la valeur de sortie de la fonction n'est pas discrète, mais continue, on parle alors de régression.

Lorsque les exemples ne sont pas étiquetés, on peut procéder à de l'apprentissage non-supervisé. Il permet de découvrir des groupes d'exemples similaires, d'analyser la distribution des exemples dans l'espace ou encore de réduire la dimension des exemples par projection dans un but de visualisation.

Il existe également d'autres formes d'apprentissage comme l'apprentissage par renforcement [SB98] qui consiste à trouver les actions appropriées à effectuer dans les bonnes situations pour maximiser une récompense.

Nous nous intéresserons par la suite principalement à la classification.

2.1.1 Processus d'apprentissage supervisé

Vocabulaire

Un exemple est une observation, un cas particulier, il appartient à un *univers d'exemples* infini, noté U . Afin de pouvoir apprendre de ces exemples, on les décrit selon un langage de description.

L'*espace des descriptions*, noté L , représente l'ensemble des descriptions qu'il est possible de former avec ce langage. L est ainsi un ensemble fini de descriptions et tout exemple dispose d'une et une seule description. Plusieurs exemples peuvent cependant avoir la même description et certaines descriptions peuvent ne correspondre à aucun exemple.

Il existe des lois générales qui attribuent une valeur cible à chaque exemple. On distingue alors les exemples d'apprentissage, étiquetés avec cette valeur et les exemples non étiquetés, dont on ne connaît pas la valeur a priori.

Le rôle de l'apprentissage supervisé est de retrouver une de ces lois générales, par généralisation des exemples d'apprentissage, pour pouvoir prédire la valeur à associer aux exemples non étiquetés, parmi un *ensemble de valeurs cibles* possibles C .

Nous appellerons ainsi *hypothèse*, une fonction de L dans C , formée par apprentissage d'une loi générale. Elle permet d'associer à tout exemple e au moyen de sa description $d(e)$, une valeur cible $c(e)$. Afin de pouvoir formuler une hypothèse, on limite leur recherche à un espace des hypothèses L_h , dépendant du langage de description des hypothèses choisi.

Nous manipulerons essentiellement des descriptions d'exemples lors de l'apprentissage. Nous identifierons un exemple à sa description.

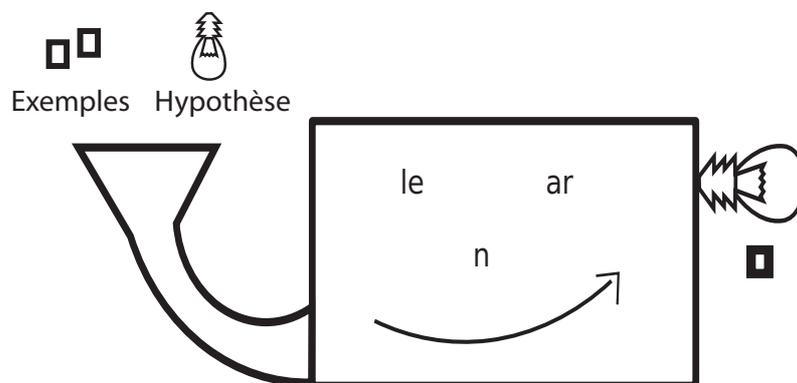


FIGURE 2.1 – Un processus d’apprentissage qui peut prendre en entrée des exemples et une hypothèse et, selon l’étape considérée, donner en sortie une nouvelle hypothèse ou une valeur cible pour un exemple non étiqueté.

Processus d’apprentissage

Le processus d’apprentissage représenté figure 2.1 est constitué de deux phases bien séparées : l’apprentissage et la restitution.

- Apprentissage
 - Apprentissage batch : Formation d’une hypothèse à partir d’un ensemble d’exemples d’apprentissage (étiquetés).
 - Apprentissage incrémental : Révision d’une hypothèse en modifiant une hypothèse d’entrée en fonction d’un ensemble d’exemples d’apprentissage.
- Restitution (Prédiction) : Utilisation d’une hypothèse pour attribuer une valeur cible à un exemple non étiqueté.

L’apprentissage peut être effectué en ligne ou hors ligne. Un apprentissage hors ligne correspond à une situation où la restitution arrive après que la collecte des données d’apprentissage soit terminée. Les données peuvent toutes être assimilées en une seule fois. Un apprentissage en ligne [Mel98] ne sépare pas autant les phases d’apprentissage et de restitution. L’apprentissage se poursuit avec une arrivée régulière de nouveaux exemples tandis que l’on souhaite pouvoir interroger le modèle à tout moment. Un apprentissage incrémental est particulièrement indiqué pour un apprentissage en ligne mais peut également être utilisé hors ligne pour diminuer le nombre de données traitées en même temps par exemple [DG01]. Un apprentissage en ligne peut

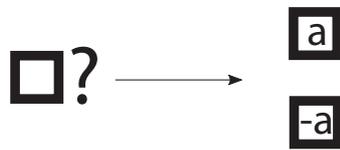


FIGURE 2.2 – L'apprentissage de concept considère deux classes, positive (le concept) et négative (hors concept).

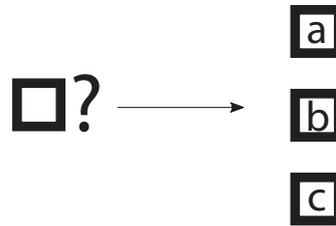


FIGURE 2.3 – L'apprentissage en classification permet de classer selon plus de deux classes.

être sujet à une dérive temporelle, c'est-à-dire que les notions à apprendre changent au cours du temps, ce qui rend les exemples les plus anciens peu fiables, mais ce n'est pas nécessairement le cas.

Apprentissage de concept et classification

En classification, une hypothèse doit permettre d'attribuer la bonne valeur à chaque exemple parmi un nombre fini de valeurs appelées classes. Une hypothèse classe les exemples selon cette valeur de sortie. L'apprentissage de concept est une sous-catégorie de l'apprentissage en classification.

L'apprentissage de concept permet de répondre à la question binaire, illustrée figure 2.2 : cet exemple appartient-il ou non au concept ? Par exemple, dans le cadre de la reconnaissance d'images, on peut chercher à apprendre le concept de chat : "est-ce un chat, oui ou non ?"

L'apprentissage en classification permet de répondre à une question admettant un plus grand nombre de réponses (figure 2.3). Les réponses possibles sont les classes envisagées. Par exemple, dans le cadre de la reconnaissance d'images, on peut chercher à apprendre la classe chat parmi d'autres : "est-ce un chat, un lapin ou une grenouille ?"

Les exemples d'apprentissages comportent, en plus de leur description de cas particulier, une étiquette indiquant la classe à laquelle ils appartiennent. Dans le cas de l'apprentissage de concept, les classes considérées sont la classe positive, si l'exemple appartient au concept, et négative s'il n'y appartient pas. Une hypothèse formée pendant la phase d'apprentissage est une fonction qui classe les exemples d'apprentissage dans la classe correspondant à leur étiquette. Cette hypothèse est utilisée en phase de restitution pour prédire

l'étiquette de nouveaux exemples qui en sont dépourvus.

Les hypothèses peuvent être des classifieurs de formes très distinctes selon l'apprentissage considéré. Il peut s'agir par exemple d'un arbre de décision [BFSO84] comme la flore Bonnier [BD11], une fonction probabiliste comme un classifieur bayésien naïf [Ris01] ou encore une fonction logique, formée d'un ensemble de règles [XYL15] ou d'une formule booléenne sous forme normale disjonctive [BES07].

Décomposition de l'apprentissage multi-classes

Mohamed Aly résume dans son article [Aly05] de nombreuses méthodes pour aborder l'apprentissage multi-classes. En particulier, il est possible de décomposer une tâche de classification en plusieurs tâches d'apprentissage de concept. Cette méthode permet ensuite d'utiliser certains algorithmes très efficaces en apprentissage de concept.

Nous présentons trois méthodes basiques pour remplacer un apprentissage multi-classes par un ensemble d'apprentissage de concept.

- Un contre tous

Cette Méthode consiste à associer un concept à chaque classe et apprendre une hypothèse pour chaque concept en considérant les exemples de la classe comme positifs et les exemples de toutes les autres classes comme négatifs [RK04]. Une prédiction peut cependant exclure un exemple de tous les concepts et il est nécessaire de rajouter un mécanisme pour les départager tel qu'une valeur de distance entre un exemple et une hypothèse. Il est possible qu'un exemple dépende de plusieurs classes, mais si les classes sont exclusives, il faut également utiliser un procédé similaire pour départager les classes prédites.

Cette méthode nécessite K concepts pour discriminer entre K classes, mais le nombre des exemples d'apprentissage de classe positive et ceux de classe négative pour chaque concept est réparti de façon très disproportionnée.

- Tous contre tous

Cette méthode définit un concept par paire de classe. On pourrait nommer ces concepts $A_plutot_que_B$, $A_plutot_que_C$, $B_plutot_que_C$... Chacun de ces concepts est appris à partir des exemples des deux classes considérées uniquement tandis que les exemples des autres classes sont ignorés [HT98]. Lors d'une phase de restitution, un vote

est effectué en donnant une voix à chacun des concept. La classe qui a le plus de voix est alors attribuée à l'exemple à classer.

Avec cette méthode, les exemples d'apprentissages sont mieux proportionnés pour chaque concept, mais il est nécessaire d'apprendre beaucoup plus de concepts, soit $K * (K - 1)/2$ concepts pour K classes.

- Error-correcting output-coding
Cette méthode présentée entre autre par Dietterich et Bakari [DB95] nécessite un nombre arbitraire de concepts N supérieur à K le nombre de classes. Chaque concept correspond à un ensemble de classes différent. On associe alors à chaque classe un mot binaire de taille N dont chaque caractère représente l'appartenance (valeur 1) ou non (valeur 0) à chacun des concepts.

En phase de restitution, chaque exemple obtient un mot binaire selon son appartenance aux différents concepts. L'exemple est attribué à la classe la plus proche, caractérisée par la distance de Hamming la plus courte, c'est-à-dire la classe dont le mot binaire présente un minimum de bits de différence avec le mot de l'exemple. Pour faciliter la classification, il est donc préférable de choisir les concepts de sorte que les distances de Hamming entre chaque paire de classes soient maximales.

Ces méthodes ont été éprouvées et améliorées de nombreuses manières différentes dont l'unification proposée par Allwein et al [ASS01].

Nous nous concentrons, dans ce document, sur l'apprentissage de concept tout en gardant à l'esprit qu'il est possible d'étendre ces méthodes à l'apprentissage en classification.

2.1.2 Apprentissage réalisable et exact

L'apprentissage peut être *réalisable*, c'est-à-dire qu'il existe nécessairement un modèle qui ne sera jamais contredit par aucune observation. Ce postulat sous-entend que tout exemple est correct. Ceci est rarement supposé pour des raisons de fiabilité tant dans la méthode d'acquisition de ces exemples (capteurs défectueux) que dans la source de cet exemple (serveur malicieux, être humain ...). Il permet cependant une simplification du problème d'apprentissage et peut donner de bons résultats malgré la présence de quelques exemples incorrects, tant qu'ils ne sont pas présents en grande proportion.

Si un exemple appartient au concept il est dit *positif*, sinon il est *négatif*. Étant donné un ensemble d'exemples E , on notera E^+ le sous-ensemble des

exemples positifs de E et E^- le sous-ensemble des exemples négatifs. L'agent forme une *hypothèse* du concept qui s'exprime comme un classifieur capable de prédire si un exemple appartient ou non au concept. On dénotera par $h \models e$ le fait que h prédise l'appartenance de l'exemple e au concept, ce que l'on appelle *couvrir* un exemple, et $h \not\models e$ le fait que h en prédise son exclusion.

Un apprentissage *exact* classe tous les exemples sans erreur et on définit alors une propriété de consistance que doit respecter une hypothèse par rapport à l'ensemble des exemples dont dispose l'agent.

Définition 1. consistance ($cons(h, E)$)

Soit E un ensemble d'exemples et h une hypothèse.

h est *consistante* avec E , noté $cons(h, E)$ si et seulement si on a :

1. $\forall e \in E^+ h \models e$
2. $\forall e \in E^- h \not\models e$

Ainsi une hypothèse consistante pour un agent par rapport à ses exemples est une hypothèse (1) *complète*, si elle couvre tous les exemples positifs et (2) *correcte*, c'est-à-dire qu'elle ne couvre aucun exemple négatif. Par exemple, figure 2.4, un agent faisant de la reconnaissance d'images, cherchant à apprendre le concept de chat, ayant un exemple négatif de grenouille et un exemple positif de chat, peut en tirer comme hypothèse consistante une règle sur la forme de la bouche. L'ajout d'un exemple supplémentaire qui contredit cette hypothèse en déclenche nécessairement sa révision. Par exemple (figure 2.4) l'ancienne hypothèse classe l'image de lapin à tort dans la catégorie chat, la révision de l'hypothèse qui tient compte de la forme de l'oreille en plus de la forme de la bouche permet de rétablir la consistance de l'hypothèse par rapport aux exemples reçus et mémorisés par l'agent.

Nous supposons par la suite un apprentissage réalisable que nous traiterons de manière exacte.

2.1.3 Apprentissage symbolique

Formulation

En apprentissage symbolique, on définit un *vocabulaire* comme étant un ensemble V constitué de *littéraux*, les *atomes* $m_1, m_2 \dots m_n$ et leurs négations

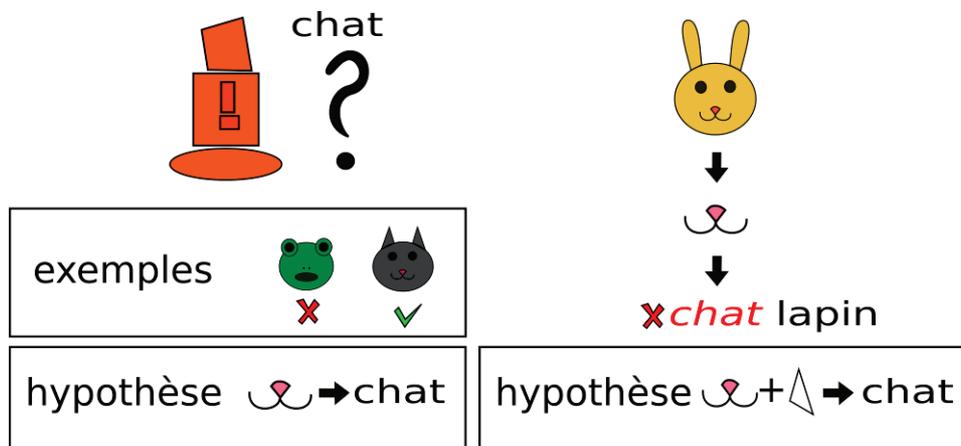


FIGURE 2.4 – Apprentissage du concept de chat à partir d’images simplifiées.

$\neg m_1, \neg m_2 \dots \neg m_n$. Les exemples s’expriment comme une conjonction d’éléments de V . Nous dirons d’un exemple qu’il est *complet* s’il est constitué d’une et une seule fois chaque atome ou sa négation.

Nous utilisons des hypothèses sous forme de prototype, c’est-à-dire une formule sous forme normale disjonctive, constituée d’une disjonction de *termes* qui sont des conjonctions d’éléments de V . Ces notions sont illustrées dans l’exemple 1. Nous définissons la *taille* d’une hypothèse comme étant le nombre de termes qu’elle contient.

Il est possible d’apprendre à partir de données incomplètes, par exemple en utilisant des connaissances a priori, comme dans les travaux de Bouthinon [BS14] concernant l’apprentissage en situation d’ambiguïté. Cette situation ne sera pas abordée en détail dans ce document, les exemples seront donc toujours complets.

Exemple 1 Un concept de chat

En considérant un vocabulaire restreint :

$$V = \{bouche3, \neg bouche3, oreillePointue, \neg oreillePointue, jaune, \neg jaune\}$$

Un exemple est de la forme :

$$bouche3 \wedge oreillePointue \wedge \neg jaune$$

Un exemple d’apprentissage est étiqueté par rapport au concept :

$$(bouche3 \wedge oreillePointue \wedge \neg jaune, positif)$$

Quelques hypothèses du concept chat :

bouche3 : taille 1

bouche3 \wedge *oreillePointue* : taille 1

oreillePointue \vee \neg *jaune* : taille 2

Pour chaque exemple e complet, on peut attribuer une valeur à chaque atome (0 ou 1 ; vrai ou faux ; positive ou négative) selon que lui ou sa négation y est présent. Si une formule booléenne h est vérifiée à partir des valeurs de vérité d'un exemple e alors elle le considère comme positif ($h \models e$), sinon elle le classe comme négatif.

Quelques problèmes booléens

Les problèmes d'apprentissage difficiles. Comme le rappellent Mason et Watts dans leur article sur l'apprentissage collaboratif dans les réseaux [MW12], les problèmes d'apprentissage difficiles nécessitent pour être traités au mieux, un compromis entre l'exploitation de ses acquis et l'exploration de l'espace des hypothèses.

Ils expriment une manière de reconnaître un problème difficile à optimiser selon plusieurs dimensions. Il faut pour cela représenter les performances des solutions selon plusieurs paramètres dans une carte multi-dimensionnelles. On reconnaît la complexité du problème à la rugosité de cette carte. Un problème simple est relativement lisse et une exploration incrémentale autour de la meilleure solution permet de converger vers la solution optimale. Un problème difficile présente beaucoup de montagnes et vallées correspondants à des optimums locaux séparés par des zones de faibles performances. L'exploration locale conduit alors vers un optimum local sans garantir l'optimalité globale. Ce format de paysage explique l'intérêt à trouver un équilibre entre une exploration locale tenant compte des résultats préalables et une autre plus distante pour découvrir des nouvelles solutions potentiellement plus performantes. On retrouve l'analogie de la difficulté d'un problème avec la rugosité d'un paysage dans d'autres travaux [Ang98] qui développent également la nécessité d'un compromis entre exploitation des résultats actuels et exploration plus lointaine.

Des problèmes booléens difficiles. Nous illustrerons l'apprentissage de concepts notamment avec l'apprentissage de formules booléennes. Nous présentons ici deux de ces formules reconnues comme difficiles, le multiplexer-11 et le ou-exclusif.

Ces problèmes sont parfois complexifiés par l'ajout de k variables supplémentaires sans influence. La nomenclature du problème est alors modifiée

ex	a_2	a_1	a_0	d_0	d_1	d_2	d_3	d_4	d_5	d_6	d_7	Val
e_1	0	0	0	1	0	0	0	0	0	0	0	+
e_1	0	0	1	1	0	0	0	0	0	0	0	-
e_2	1	1	1	1	0	0	1	1	0	0	1	+
e_3	1	1	1	1	0	0	1	1	0	0	0	-

TABLE 2.1 – Quelques exemples positifs et négatifs du problème booléen M11.

par l'ajout de $_k$ à la fin de leur nom.

- Multiplexer-11 ($M11$)

Le M11 est une formule booléenne sur 11 variables. Elle est vérifiée si les trois premières variables a_0, a_1, a_2 prennent pour valeur binaire un nombre entre 0 et 7, correspondant à une des huit autres variables ($d_0 \dots d_7$) dont la valeur est "vrai". Un exemple $e_2 = a_0 \wedge a_1 \wedge a_2 \wedge d_0 \wedge \neg d_1 \wedge \neg d_2 \wedge d_3 \wedge d_4 \wedge \neg d_5 \wedge \neg d_6 \wedge d_7$ appartient bien au concept puisque les termes a_0, a_1, a_2 sont présents, on leur associe la valeur (1,1,1) soit le nombre 7, et d_7 est bien également présent. On donne quelques autres exemples dans la table 2.1.

- ou-exclusif (XOR_a)

Les formules booléennes XOR_a indiquent que parmi a variables, seul un nombre impair d'entre elles doivent être vraies. Pour le XOR_{5_2} , chaque exemple est constitué de 5 variables (d_0, d_1, d_2, d_3, d_4) qui doivent être présentes en nombre impaires et deux variables (i_0, i_1) sans influence. L'exemple $e = d_0 \wedge \neg d_1 \wedge \neg d_2 \wedge d_3 \wedge d_4 \wedge i_0 \wedge \neg i_1$ comporte 3 des 5 variables importantes, il est donc positif.

Ces deux problèmes ont comme propriété de disposer d'autant d'exemples positifs que d'exemples négatifs. Une hypothèse vide, qui rejette tous les exemples, en classe donc correctement la moitié. Le paysage de la probabilité d'erreur dans l'espace des hypothèses présente des optimums locaux ce qui en fait des problèmes difficiles nécessitant de faire un compromis entre exploration et exploitation pour en apprendre la formule.

Esmeir et Markovich [EM04] abordent de tels problèmes booléens difficiles lors de l'apprentissage d'arbres de décision. Ils étudient plusieurs algorithmes d'apprentissage qui permettent de trouver une bonne solution rapidement, mais surtout poursuivent son amélioration tant qu'ils ne sont pas interrompus. La recherche d'amélioration se base sur une exploration plus ou moins lointaine autour de la solution actuelle.

Choix de l'apprentissage symbolique

Il existe de nombreuses techniques d'apprentissage non symbolique : support vecteur machine [Vap95], arbre de décision [BFSO84, QR89], réseau de neurones [Bis95], réseau bayésien [Pea88], k plus proches voisins [DH73, DK82]...

Nous nous intéressons cependant plus à une simulation de comportements sociaux dans laquelle on puisse distinguer des individus et des phénomènes de groupe. L'apprentissage symbolique est donc plus indiqué, il permet de suivre les informations et les raisonnements et le résultat de l'apprentissage peut s'exprimer comme une règle traduisible dans un langage proche du nôtre.

2.1.4 Algorithme d'apprentissage incrémental ascendant

Dans la suite de ce travail, nous utiliserons un processus d'apprentissage basé sur le principe de plus petite généralisation, l'algorithme Imind2 présenté par Henniche [Hen97]. Il s'agit d'un algorithme d'apprentissage de concept symbolique incrémental et exact utilisant une méthode dite ascendante (ou "bottom-up").

Dans ce cadre, une hypothèse est une disjonction de termes et les exemples sont décrits par des termes complets. Les termes sont une conjonction d'atomes d'un vocabulaire V . L'ensemble des exemples qui peuvent être formés à partir du vocabulaire V est noté $M\{V\}$.

On définit ici la notion de couverture entre des termes : un terme t_1 couvre un terme t_2 (noté $t_1 \models t_2$) ssi tous les atomes de t_1 apparaissent dans t_2 . Cette notion de couverture s'étend aux hypothèses en considérant qu'une hypothèse h couvre un terme t ssi l'un de ses termes couvre t . Appliquée aux exemples, cette définition correspond à celle donnée plus généralement en section 2.1.2.

Tout terme t formé à partir de V couvre un sous ensemble de $M\{V\}$ noté C_t^V qui vérifie :

$$\left\{ \begin{array}{l} C_t^V \subset M\{V\} \\ \forall e \in C_t^V, t \models e \\ \forall e \notin C_t^V, t \not\models e \end{array} \right.$$

La *plus petite généralisation* de deux termes t_1 et t_2 , notée $lgg(t_1, t_2)$ est un terme couvrant un sous-ensemble de $M\{V\}$ de taille minimum et qui couvre au moins les deux sous-ensembles d'exemples couverts par t_1 et par t_2 .

Définition 2. Plus petite généralisation (lgg - least general generalisation)

On définit la plus petite généralisation des termes t_1 et t_2 , comme le terme $lgg(t_1, t_2)$ qui vérifie :

$$\left\{ \begin{array}{l} (i) C_{t_1}^V \subset C_{lgg(t_1, t_2)}^V \\ (ii) C_{t_2}^V \subset C_{lgg(t_1, t_2)}^V \\ (iii) \forall t, (C_{t_1}^V \subset C_t^V, C_{t_2}^V \subset C_t^V) \Rightarrow \text{card}(C_t^V) \geq \text{card}(C_{lgg(t_1, t_2)}^V) \end{array} \right.$$

Dans le cas présent, $lgg(t_1, t_2)$ est le terme formé de la conjonction de tous les atomes présents à la fois dans le terme t_1 et dans le terme t_2 .

Lorsqu'un agent acquiert un contre-exemple, son hypothèse perd sa propriété de consistance. Si l'exemple est positif, l'hypothèse n'est plus complète, tandis que si l'exemple est négatif, l'hypothèse n'est plus correcte. Ces deux cas sont donc traités différemment :

Algorithme Imind2 :

Soit un nouvel exemple e ajouté à la mémoire E de l'agent ayant une hypothèse $h = h_{t1} \vee h_{t2} \dots \vee h_{tn}$:

1. Si $\text{cons}(h, e)$ alors Fin
 Si e n'est pas un contre-exemple, h reste consistante avec E , elle n'est donc pas révisée.
2. Si $\text{not cons}(h, e)$ et $e \in E^+$ cas positif
 si $\exists k$ tel que $\forall e^- \in E^-, lgg(h_{tk}, e) \not\subseteq e^-$
 alors $h' = h_{t1} \vee \dots \vee h_{tk-1} \vee lgg(h_{tk}, e) \vee h_{tk+1} \dots \vee h_{tn}$
 sinon $h' = h_{t1} \vee \dots \vee h_{tn} \vee e$

e n'est pas couvert par h ($h \not\subseteq e$), la révision de h en h' doit couvrir e tout en restant consistante avec les autres exemples. On cherche tour à tour à généraliser les termes h_t de h pour qu'ils couvrent e sans couvrir d'exemple négatif. Si, pour un des termes h_{tk} de h , $lgg(h_{tk}, e)$ ne couvre aucun exemple négatif alors $lgg(h_{tk}, e)$ remplace le terme h_{tk} dans h pour former h' . S'il n'est pas possible de généraliser aucun des termes de h sans couvrir d'exemple négatif on ajoute le terme supplémentaire e à h pour former h' .

3. Si $\text{not cons}(h, e)$ et $e \in E^-$ cas négatif

$$h' = \bigvee \{h_{ti} | h_{ti} \not\models e, i \in \{1..n\}\}$$

$\forall e^+ \in E^+$ tel que $h \models e^+$ et $h' \models e^+$ réviser h' selon 2

Si e est couvert par un terme t de h , il n'est pas possible de le rejeter par généralisation, il faut donc retirer ce terme t pour former h' . Les exemples positifs qui étaient exclusivement couverts par les termes retirés doivent de nouveau être couverts par h' . L'ensemble des exemples positifs à couvrir de nouveaux E_p vont être recouverts par différentes tentatives de spécialisation du terme t . Cette spécialisation consiste à ajouter à t un atome absent du contre-exemple e afin que ce nouveau terme ne couvre plus e . Une spécialisation de t en t' sera retenue si elle couvre suffisamment d'éléments de E_p . L'hypothèse h' à laquelle on ajoute le terme t' est alors révisée selon l'étape 2 pour tous les exemples de E_p restant à couvrir de nouveau.

Enfin l'hypothèse est réduite en retirant tout terme qui soit couvert par un autre de ses termes.

Ce processus de révision garantit le maintien de la consistance de l'hypothèse à chaque nouvelle arrivée d'exemple.

2.1.5 Agents apprenants

Cas général

Le terme d'agent est souvent utilisé en apprentissage. On peut utiliser la définition de Ferber [Fer89] :

“Un agent est une entité (physique ou virtuelle) capable d'agir sur elle-même et sur son environnement, disposant d'une représentation partielle de cet environnement, pouvant communiquer avec d'autres agents et dont le comportement est la conséquence de ses objectifs, de sa perception, de ses représentations, de ses compétences et des communications qu'elle peut avoir avec les autres agents.”

Un agent est souvent un programme informatique ou une fonction capable de fournir une sortie dépendant de données d'entrée. Dans le cadre de l'apprentissage, on suppose qu'il dispose d'un processus d'apprentissage lui permettant de fournir la fonction qu'il a apprise ou de s'en servir pour attribuer une valeur à un exemple non étiqueté. On souhaite que ces agents puissent représenter toute sortes de capteurs intelligents, robots ou individus au sein d'un réseau social.

Agent social

Afin de pouvoir considérer un apprentissage collectif, nécessairement multi-agents, il est nécessaire de considérer qu'un agent peut interagir avec d'autres. Les agents qui ne sont pas isolés ont des moyens de communication.

Agent autonome

Un concept important pour l'apprentissage collectif est l'autonomie des agents. Une manière de considérer des agents comme autonomes est de leur attribuer des intentions ou bien une indépendance d'action.

Ceci peut être envisagé avec une architecture BDI (Belief Desir Intention) [RG91, GHEFSS05] qui distingue croyance désir et intention. Les croyances sont la partie des connaissances qui sont incertaines. Les désirs d'un agent doté d'une telle architecture peuvent être multiples et incohérents, c'est pourquoi il a aussi des intentions. Ce sont les désirs, cohérents entre eux, qu'il a choisi de réaliser.

Un agent dispose d'une indépendance d'action dans le cadre de l'apprentissage actif [Set12]. Cela consiste à donner la possibilité à un agent de planifier et réaliser des protocoles expérimentaux. Il décide ainsi des prochains exemples d'apprentissage qu'il va acquérir, selon sa propre estimation de ses besoins.

Bien qu'il soit possible d'y étendre la présente étude, il n'est pas nécessaire d'aller jusqu'à un niveau de représentation de l'agent BDI ni au potentiel de l'apprentissage actif pour conférer de l'autonomie aux agents. Il s'agit seulement de leur permettre d'apprendre indifféremment seul ou en groupe. Un agent autonome doit alors pouvoir être déconnecté des autres sans rendre la poursuite de son apprentissage ni sa restitution actuelle impossible.

Facultés d'un agent

Un agent est défini plus précisément par les facultés qu'on lui attribue. Nous utiliserons, dans le cadre précis de cette étude, à partir du chapitre 3, la définition suivante d'un agent :

On appelle *agent*, comme schématisé figure 2.5, un programme capable de percevoir des exemples, de les mémoriser et d'apprendre à prédire s'ils appartiennent ou non au concept à apprendre. Il dispose, de plus, de facultés de communication avec les autres agents.

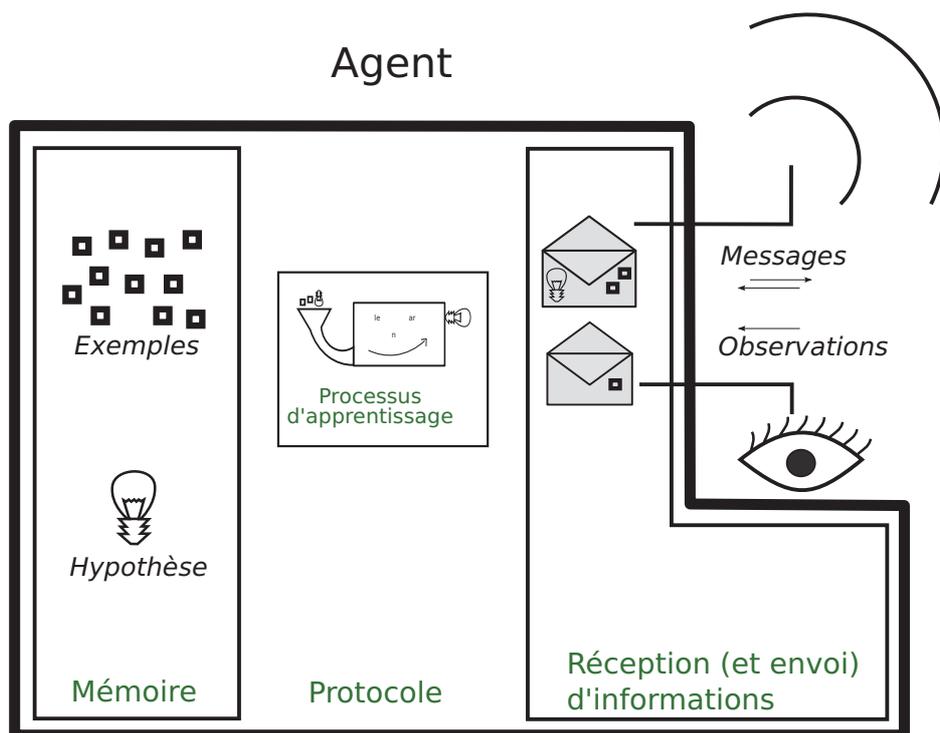


FIGURE 2.5 – Représentation schématique d'un agent. Ses principales capacités sont de recevoir des exemples, les mémoriser et pouvoir former et réviser des hypothèses par un processus d'apprentissage. Au sein d'un groupe il aura également une faculté de communication par envoi et réception de messages pouvant contenir une hypothèse ou des exemples.

2.1.6 Apprentissage par requêtes

On s'inspire pour divers protocoles multi-agents de l'apprentissage par requête (query learning).

Le principe de l'apprentissage par requête est de permettre, à un apprenti, de poser les questions qu'il souhaite à un oracle omniscient. Dans la mesure où cet oracle est l'environnement et qu'il est toujours disponible, l'apprenti peut donc obtenir de nouvelles données de son choix pour nourrir son apprentissage. Tel qu'il est présenté par Angluin [Ang88][Ang04], un apprenti peut formuler diverses requêtes à des oracles concernant un concept à apprendre dans un espace d'hypothèses fini et dénombrable. Une hypothèse H est assimilée à l'ensemble des descriptions d'exemples qu'elle couvre. Un apprenti cherche, dans l'espace des hypothèses L_h , une hypothèse L_* qui couvre le même sous-ensemble de descriptions d'exemples que le concept à apprendre.

On distingue deux principales formes de requêtes, la requête d'appartenance d'un exemple au concept et des requêtes de comparaison entre les ensembles d'exemples représentés par une hypothèse et par le concept cible. La comparaison la plus utilisée est la requête d'équivalence. Ces requêtes sont définies comme suit :

- *appartenance*. La question porte sur un exemple $x \in U$ et sur son appartenance au concept à apprendre L_* . La réponse obtenue est de la forme *oui* ou *non* respectivement si $d(x) \in L_*$ ou $d(x) \notin L_*$
- *équivalence*. La question porte cette fois sur une hypothèse H et son équivalence avec le concept à apprendre L_* . La réponse est *oui* si $H = L_*$ et *non* si $H \neq L_*$. Si la réponse est non, un contre exemple ce est également renvoyé à l'apprenti. ce est choisi tel que $d(ce) \in H \oplus L_*$, c'est-à-dire que sa description est couverte par un et un seul des deux ensembles parmi H et L_* .

Il existe d'autres formes de comparaisons que l'équivalence. Il est possible de faire une requête demandant si l'hypothèse est incluse dans le concept, si elle l'inclut, si elle est exhaustive, ou encore si les deux ensembles sont disjoints. De manière intéressante, les réponses aux requêtes de comparaisons sont de la forme "oui ou non", auxquelles sont ajouté un contre-exemple dans le cas d'une réponse négative. Pour la requête d'appartenance, l'exemple de la question peut lui-même servir de contre-exemple, que la réponse de l'oracle soit positive ou négative. Dans tous les autres cas, une nouvelle donnée est fournie à l'apprenti lors d'une réponse négative.

L'apprentissage actif correspond à des requêtes d'appartenance d'un exemple à un concept, tandis que nous nous intéresserons plutôt aux requêtes d'équivalence fournissant un contre-exemple.

En supposant qu'il n'y a pas d'oracle omniscient disponible pour répondre à de telles requêtes, un apprenti peut se tourner vers les oracles imparfaits que sont l'ensemble des autres apprentis. Ceux-ci ayant acquis des connaissances potentiellement différentes, le mécanisme de requêtes et contre-exemples peut nourrir les connaissances de l'apprenti. Nous verrons comment ce mécanisme est mis en place dans l'apprentissage collectif.

2.2 Apprentissage collaboratif

L'apprentissage collectif est une sous-catégorie de l'apprentissage multi-agents collaboratifs.

Les agents ne sont plus isolés mais réunis au sein d'un réseau représenté par un graphe dont les agents sont des sommets et les arêtes représente la capacité à communiquer. Cette section s'intéresse aux différentes formes d'interactions entre agents tandis que la section suivante 2.3 développe les caractéristiques dynamiques et l'influence de la structure du réseau. La partie 2.2.1 présente des caractéristiques très générales concernant l'apprentissage multi-agents tandis que la partie 2.2.2 montre que la caractéristique collaborative se réfère à des formes d'apprentissage encore très variées. Les parties suivantes détaillent différentes catégories d'apprentissage collaboratifs avant d'être conclues en 2.2.8 par l'apprentissage collectif.

2.2.1 Identifier une forme d'apprentissage multi-agents

Du point de vue de l'intelligence artificiel, on distingue plusieurs catégories d'apprentissage multi-agents. Pour une même tâche d'apprentissage, les agents, réunis dans un même système, agissent de manière plus ou moins indépendante.

Centralisé ou décentralisé. La tendance prédominante en apprentissage artificielle vise à centraliser les données comme avec un protocole d'agrégation (figure 2.6b) avant d'apprendre de celles-ci. Les progrès en big data permettent de traiter de tels flux de données de très grande taille. On peut cependant considérer pour des raisons de protection de données personnelles ou de sécurité que l'on ne peut pas ou ne souhaite pas centraliser ces données.

Les données sont donc acquises et mémorisées séparément dans plusieurs lieux.

L'Adversité. Beaucoup utilisé en théorie des jeux, l'apprentissage dans l'adversité consiste plus à apprendre à s'adapter à un comportement (figure 2.6c) qu'à apprendre un concept. Ainsi, certains jeux s'adaptent à leurs joueurs pour leur offrir un défi à leur niveau [BSvL12]. Des jeux qui demandent plus qu'une maîtrise des règles et une évaluation de l'ensemble des coups possibles disposent d'une intelligence artificielle capable de dépasser les performances humaines, comme le montre l'exemple emblématique d'alphaGo [SHM⁺16] pour le jeu de Go. L'adversité dépasse le cadre du jeu et concerne des domaines de progrès compétitifs comme la sécurité ou le filtrage de spam [ZKTX12].

En l'absence d'adversité, la présence d'autres agents, ne modifie pas le concept cible. Lorsque les agents partagent leur environnement, il est encore possible cependant qu'il y ait une modification des données d'apprentissages ou de leur disponibilité par la simple présence d'autres agents.

La Division du travail. La division du travail peut avoir lieu lors de la phase d'apprentissage, elle implique alors la spécialisation de chaque agent. C'est une division du travail en plus petites tâches simples. Les hommes ont appliqué cette méthode de travail à la chaîne dans le fordisme, mais elle montre de meilleurs résultats en IA, comme par exemple avec les réseaux de neurones, les programmes étant adaptés à effectuer des tâches répétitives.

La division du travail peut également se situer dans la restitution (figure 2.6d) en apprenant une technique qui ne peut être effectuée qu'à plusieurs. Les agents peuvent apprendre à agir ensemble comme par exemple deux robots se positionnant de part et d'autre d'un objet pour le soulever et le déplacer.

Un apprentissage collectif (figure 2.6e) correspond à un cas d'apprentissage décentralisé, sans adversité et qui est motivé par l'autonomie de ses individus plutôt que par une division du travail.

2.2.2 La diversité des apprentissages collaboratifs

Entre coopération et bénéfice

Le terme collaboration est utilisé dans de nombreux domaines avec un sens et des applications parfois assez éloignées comme le souligne Dillenbourg dans le chapitre introductif de son livre [Dil99a] co-écrit par des spécialistes

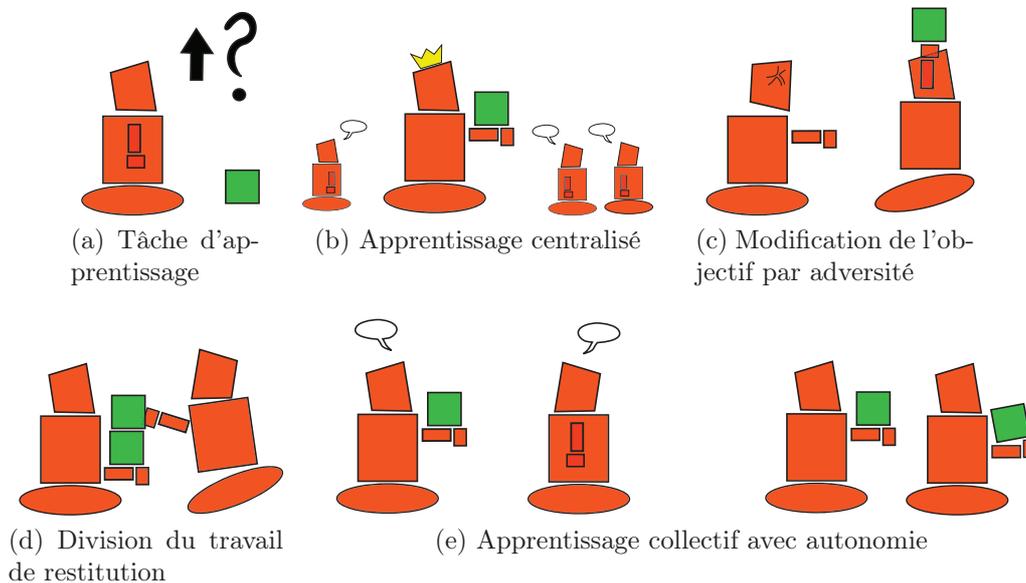


FIGURE 2.6 – Différentes possibilités d'interactions multi-agents

en psychologie cognitive, science de l'éducation et intelligence artificielle. Il propose une définition très large d'apprentissage collaboratif :

"[...] is a situation in which two or more people learn or attempt to learn something together."

Une autre définition de la collaboration souvent prise comme référence est celle de Roschelle et Teasley [RT95] (p70) :

"Collaboration is a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem. "

Nous distinguons ici deux composantes principales, ce que nous appellerons la *coopération* des acteurs d'une part et l'intérêt personnel c'est-à-dire le *bénéfice* de chacun d'autre part.

Nous considérons qu'il existe différents niveaux de coopération. La coopération englobe à la fois le nombre de partenaires de chaque acteur, l'activité avec laquelle il participe à la coordination du travail commun et la complexité des interactions mises en jeu. Ainsi une coopération nulle concerne des agents isolés. Une coopération faible peut être soit la mise en commun de données personnelles sans aucune interaction entre utilisateurs comme on

le retrouve dans le domaine des recommandations avec le filtrage collaboratif [RIS⁺94, HB05, PZ01], soit une architecture très centralisée et d'apprentissage distribué. Dans ce deuxième cas, des données centralisées sont découpées en sous-ensembles et distribuées à différents nœuds de calculs qui en font l'analyse. Les résultats sont alors centralisés de nouveau pour être exploités. Un cas extrême comme Hadoop [IJL⁺09] devient une technique de calcul en parallèle, ici le MapReduce, plutôt qu'une situation d'apprentissage collaboratif. Le traitement en parallèle permet d'accélérer les temps de calculs, ce qui rend possible le traitement de grosses quantités de données et de diminuer la latence. La relation maître-esclave ainsi que l'absence de communication entre les nœuds de calculs et même la nature des interactions qu'ils ont avec l'agent central (recevoir un ensemble de données, envoyer un résultat) caractérisent un niveau de coopération faible. L'intelligence artificielle distribuée présente également des techniques plus coopératives pour tenir compte des résultats individuels depuis le simple vote [Bre96] jusqu'à des techniques avancées d'argumentation [XYL15]. De plus, certains algorithmes locaux favorisent les interactions entre agents pour une plus grande coopération [WBK09, VBS17b].

Le bénéfice d'un apprentissage peut être évalué au niveau collectif mais aussi au niveau individuel. Les deux définitions de collaboration données ci-avant insistent également sur ce point, le résultat d'un apprentissage collaboratif est partagé. Le gain de l'apprentissage peut être très inégalement réparti, dans des situations centralisées par exemple [XYL15]. Ce gain peut ensuite être redistribué aux agents dans un processus incrémental pour améliorer des résultats successifs envoyés à un agent central. Il est également redistribué lorsque l'utilisateur qui consulte ou reçoit les résultats de l'apprentissage centralisé est aussi un des agents ayant fourni les données d'origine. C'est le cas des systèmes de recommandations qui tiennent compte des votes des utilisateurs [PZ01, BS97, VYL⁺17] ou encore par exemple du système MineFleet [KGP⁺10] qui est une application commerciale de gestion de camions dont les pannes et comportements des conducteurs peuvent être discutés après confrontation des données individuelles entre elles. Parfois il n'est pas évident que les utilisateurs du résultat soient les mêmes personnes que celles qui fournissent les données, mais on peut le supposer de par leur intérêt pour le domaine. Par exemple, TagLearner [DZM⁺09] est un classifieur pour des documents texte formé à partir d'un système de tags fournis par une communauté d'individus ; ou encore PADMINI [MBD⁺10] permet la détection de données aberrantes pouvant conduire à des découvertes d'envergure à partir de données astronomiques récoltées par différents observatoires. Le gain personnel peut également être recherché activement par chaque agent

à tout moment. Le bénéfice peut prendre la forme d'une position, dans le cas de la détection d'une source de pollution par montée de gradient par exemple [COH09], d'un score personnel, comme pour un jeu d'exploration permettant de consulter les positions et scores de ses voisins [MW12], de la compétitivité d'une entreprise [KW99] ou encore d'un modèle de concept personnel tenant compte des données collectées par un grand nombre d'autres agents [BES07, OP14, VBS17b].

La collaboration couvre divers degrés, positifs ou nuls, de coopération ou de bénéfice personnel à partir d'une mise en commun d'informations ou de moyens. En considérant à la fois une forte coopération et un bénéfice personnel, nous voyons apparaître l'individualité. On parle d'*autonomie* lorsqu'un agent est capable d'apprendre seul. La coopération et le bénéfice individuel ne sont pas opposés et, lorsque les acteurs disposent d'une certaine autonomie, comme par exemple dans une structure pair-à-pair auto-organisée, on voit apparaître des problématiques dépendant de ces deux composantes.

Définition 3. L'apprentissage d'agents autonomes coopératifs qui peuvent tirer un bénéfice collectif et individuel de cette collaboration est ici ce que l'on appelle *apprentissage collectif*.

Autres caractéristiques différenciant des apprentissages variés

L'informatique n'est pas le seul domaine à porter de l'attention à l'apprentissage collaboratif. Des approches psychologiques et sociales s'intéressent tout particulièrement aux conditions mais également aux processus qui conduisent aux changements et progrès d'un apprenti [Ros92], souvent un enfant [Kuh72, CL87, RR88], lorsqu'il est placé dans un contexte collaboratif plutôt qu'isolé. Dans un contexte économique, les principales questions se posent concernant la réussite de groupe régionaux d'entreprises. On parle par exemple d'apprentissage collectif en milieu innovateur [Cam91]. Dans le domaine économique, le bénéfice de chacun est très important et la coopération doit trouver un équilibre avec la compétition.

La nature des agents considérés n'est pas toujours la même et d'autres contraintes peuvent apparaître qui modifient le problème d'apprentissage étudié. Une analyse psychologique ou sociale doit notamment tenir compte d'une dimension affective et émotionnelle ainsi que d'un contexte culturel. Une entreprise sera contrainte par une nécessité de rentabilité et le bénéfice de chacun est très important, rendant inévitable la présence de compétition. Mais même en se restreignant à l'informatique, les agents sont soumis à des contraintes dépendant de leur déploiement. Des techniques de big data nécessitent des machines spécifiques, chères et performantes. Un réseau de capteurs

sans fils (Wireless Sensor Network : WSN) est constitué d'une multitude de capteurs disposant d'une mémoire et une énergie très limitées. Des agents mobiles comme des véhicules ou des robots peuvent voir leurs communications limitées par la distance ou le cout des communications en terme d'argent ou d'énergie. Des réseaux pair-à-pair (Peer-to-peer : P2P) sont également concernés par des connexions et déconnexions. Les réseaux P2P manipulent également souvent des données personnelles et la protection de la vie privée limite les données qui peuvent être partagées.

Dans certains cas, seul le résultat compte mais on souhaite parfois pouvoir retracer le raisonnement ou expliquer ce résultat, par exemple lors de recommandations.

On comprend donc qu'il y a de nombreuses approches ainsi que de nombreuses méthodes pour réaliser de l'apprentissage collaboratif bien que peu d'entre elles relèvent de l'apprentissage collectif.

2.2.3 Apprentissage distribué et fouille de données

L'apprentissage distribué considère plusieurs algorithmes travaillant sur les mêmes données ou sur différents sous-ensembles de ces données qui ont ou vont tenter d'échanger et exploiter leurs résultats respectifs. On distingue alors généralement deux phases. La première phase consiste à produire des résultats individuels à partir des données locales. La seconde est un échange collaboratif des résultats locaux.

La coopération reste très limitée si les résultats locaux sont simplement rassemblés pour un traitement centralisé, c'est le cas des méthodes d'ensemble classiques tels que Bagging [Bre96], Boosting [Sch03] ou Random Forest [LW⁺02]. Il s'agit dans ce cas d'utiliser un vote, potentiellement pondéré, de plusieurs apprentis distincts, chacun étant légèrement plus performant qu'un choix aléatoire. Une prédiction de groupe, résultat du vote de l'ensemble des apprentis, est reconnue comme étant plus précise en moyenne que celle de chacun des apprentis pris individuellement [Bro11].

La phase d'échange peut être plus complexe qu'un simple vote comme par exemple dans les travaux de Xu [XYL15] qui utilise un procédé argumentatif entre classifieurs pour en former un global plus performant.

Dans le cas d'un apprentissage non supervisé il existe également des méthodes d'ensemble appelées cluster-ensembles [SG02] qui combinent le résultat de plusieurs apprentis sans connaître a priori l'algorithme d'apprentissage de chacun d'eux.

De nombreuses problématiques peuvent s'ajouter à celle d'un apprentissage distribué. On pourra s'intéresser par exemple à l'apprentissage d'un

concept qui n'est pas fixe dans le temps [Kun] ou aux limitations des agents apprenants, du fait de leur structure physique dans un réseau de capteurs sans fil par exemple [BS13]. On peut avoir à traiter un découpage vertical des données [KBHJ99], c'est-à-dire des données qui peuvent être observées par plusieurs sources différentes mais pas selon les mêmes attributs, au lieu d'une répartition horizontale, où les exemples sont homogènes, plus communément étudiée.

Deux autres problématiques se rejoignent et nous intéressent plus particulièrement, l'apprentissage asynchrone dans des réseaux pair-à-pair, qui empêchent de centraliser l'apprentissage, et la volonté de faire profiter, plutôt que simplement participer, tous les apprentis de la phase d'échange collaborative.

Datta et al [DBG⁺06] présentent dans le cadre de la fouille de données distribuée les enjeux et les solutions envisagées dans un réseau pair-à-pair, par nature asynchrone et dont des membres du groupe peuvent à tout moment être ajoutés ou retirés. Il s'agit souvent d'algorithmes locaux, tels qu'il existe une constante k indépendante de la taille du réseau représentant le nombre de communications maximal (ou autre ressource que l'on souhaite limiter) par pair. Ce type d'algorithme est très apprécié pour les réseaux de grande échelle mais il n'est pas toujours possible d'en faire des algorithmes aussi précis qu'un apprentissage centralisant les données. Wolf et al [WBK09] proposent un algorithme local générique qui permet de traiter indifféremment plusieurs problèmes de fouille de données.

Lorsque l'objectif de la phase collaborative est d'améliorer le modèle de chaque apprenti plutôt que d'en extraire un modèle global en apprentissage non supervisé, on parle alors de clustering collaboratif. Dans leur article, Sublime et al [SMC⁺17] proposent une méthode telle que les données n'ont pas besoin d'être partagées entre les différentes sources et les algorithmes locaux ainsi que le nombre de clusters obtenus individuellement peut être différent pour les différents membres du groupe. On peut également considérer qu'il y a un bénéfice personnel pour les agents apprentis lorsqu'ils ne font pas que donner leurs résultats mais reçoivent et utilisent également ceux des autres.

Provost et Hennessy [PH96] présentent une tâche d'apprentissage de règles satisfaisant un critère d'évaluation fixé par l'utilisateur. Ils proposent d'enrichir la coopération des agents au-delà d'une simple confirmation de la validité globale de règles obtenues localement. En partageant, pendant toute la phase d'apprentissage, des résultats partiels ils peuvent réduire leur champ de recherche et bénéficier de versions améliorées de leurs règles actuelles obtenues par d'autres agents.

Dans une autre approche, Davies et Edwards [DE95] distinguent le raffinement de théorie, qui consiste à réviser une théorie en tenant compte d'un nouvel ensemble d'exemples, de l'intégration de connaissance, qui consiste à fusionner plusieurs modèles en un seul. Ils comparent alors plusieurs manières de tenir compte des modèles locaux que chaque agent a produit à partir d'un ensemble de données homogènes mais distinctes. Toutes les méthodes envisagées ne permettent aucun échange de données brutes mais seulement du modèle résultat, ce qui est une contrainte forte de protection des données privées. Une première méthode évalue les résultats de chaque modèle sur chaque ensemble de données pour ne conserver que le meilleur. Une autre méthode effectue la transmission d'un modèle unique, d'un agent au suivant, selon une ligne et que chaque agent révisé, si besoin, conformément à ses données personnelles. Enfin, une dernière approche permet à tous les agents de recevoir l'ensemble des modèles des autres, qu'ils révisent tous individuellement, avant de procéder à une sélection, comme dans le premier cas.

Des méthodes similaires seront utilisées pour la conception de Waves [VBS17b], présenté dans cette thèse. Waves ajoute cependant une dimension collective de sorte que chaque agent souhaite avoir son propre modèle. Il permet à une succession de révisions incrémentales de converger au prix d'un envoi de certaines données brutes, des exemples, en plus de l'envoi de modèles.

En apprentissage artificiel et fouille de données, la distribution des données est parfois une condition du problème et plus souvent un moyen d'apprendre un modèle central unique en allégeant les calculs et la mémoire utilisée par agent. Les agents considérés sont donc essentiellement des outils plutôt que des acteurs indépendants. Les raisons et les méthodes de partitionnement de données autant que d'agrégation des résultats locaux sont très nombreuses pour répondre à des problèmes variés. À l'inverse, il est rare que l'apprentissage bénéficie à tous les agents participants.

2.2.4 Collaborations en recommandations

Le problème d'apprentissage rencontré dans les tâches de recommandation consiste à proposer à un utilisateur de nouveaux objets sur la base d'un ensemble d'objets qu'il a préalablement évalué. Il est plus souvent appelé recommandation collaborative ou filtrage collaboratif mais parfois le terme d'apprentissage collaboratif [HB05] apparaît également pour désigner ce problème d'apprentissage. C'est un domaine riche et actif qui offre un point de vue sur la collaboration dans l'apprentissage très différent des autres.

La recommandation peut être adaptée à de nombreux types d'objets

comme les pages internet [BS97], la musique [Cel10], les films [PZ01], les mangas [VYL⁺17], les marchandises [LSY03] ... Il existe deux grandes approches pour générer des recommandations, le filtrage basé sur le contenu [Lan95] et le filtrage dit collaboratif, basé sur les notes des autres utilisateurs [SM95], ainsi que diverses méthodes dites hybrides qui combinent ces deux approches [BS97]. Dans tous les cas il est important que l'utilisateur donne un certain nombre de notes pour pouvoir conclure sur des centres d'intérêt ou un profil d'utilisateur. Cette tâche étant laborieuse pour l'utilisateur, elle vise à être limitée par les différents algorithmes existants.

Le filtrage basé sur le contenu utilise un ensemble de descripteurs associés à l'objet pour trouver les objets similaires à recommander. NewsWeeder [Lan95], par exemple, filtre l'actualité sur internet en utilisant les votes des utilisateurs qui indiquent leur intérêt et le contenu des articles. Cette approche a des limites bien connues. Elle dépend très fortement des informations que l'on peut extraire d'un objet, or les méthodes d'extraction ne sont pas les mêmes, ni aussi développées, pour tous les types d'objets. Une page internet ou un roman peuvent être riches en texte et sont donc recommandés plus efficacement et depuis plus longtemps que la musique [Cel10]. On peut également bénéficier de tags ou annotations par l'utilisateur pour enrichir le contenu d'un objet [DZM⁺09] mais il est possible que les utilisateurs ne s'accordent pas sur les tags à attribuer et ils sont limités sur la quantité d'information qu'ils peuvent produire. Une autre limitation du filtrage de contenu est qu'il dépend beaucoup des notes précédentes de l'utilisateur et tend à se sur-spécialiser rendant impossible la recommandation de nouvelles catégories d'objets.

Le filtrage collaboratif utilise les votes d'un utilisateur pour trouver des utilisateurs avec lesquels il aurait une grande affinité et recommande ce que les personnes de goûts similaires ont apprécié d'autre. GroupLens [RIS⁺94] est un exemple de filtrage collaboratif d'actualité sur internet. Les limites de cette approche concernent essentiellement les objets peu notés et le nombre d'utilisateurs. Un nouvel objet qui n'a pas encore été noté par aucun utilisateur ne peut pas être recommandé. De même une personne au profil atypique ne trouvera pas d'utilisateurs semblables et recevra des recommandations peu adaptées.

Les approches hybrides [BS97, PZ01, HB05] tentent d'utiliser les points forts de ces deux approches en évitant leurs défauts. Les méthodes peuvent encore se diversifier pour s'adapter au type de recommandation souhaitée. Fab [BS97] qui recommande des pages internet utilise en réalité des objets

intermédiaires sous la forme d'agents collectant du contenu sur internet associé à un sujet spécifique. Ce sont les sujets qui sont évalués par l'utilisateur et associés à leur profil tandis que les agents collecteurs apparaissent ou disparaissent, selon l'intérêt porté par l'ensemble des utilisateurs à leur sujet. Ce modèle correspond à un contenu à grande échelle et évoluant rapidement que représente internet. Cette méthode envisage également d'utiliser les données personnelles des utilisateurs (âge, genre, ...) pour affiner le système de recommandation tandis que dans d'autres circonstances ce type de données est exclu. Film-Conseil [PZ01] est un système de recommandation de films dans ce cas. Il a la particularité de vouloir pouvoir fournir une explication pour une recommandation, il calcule donc indépendamment une évaluation d'un objet par filtrage de contenu et par filtrage collaboratif. Un coefficient peut ensuite être appliqué aux deux évaluations selon le nombre d'utilisateurs pour fournir une recommandation finale qui dépend d'autant plus des utilisateurs qu'ils sont nombreux et actifs. D'autres méthodes lient beaucoup plus fortement les données provenant des objets similaires et des utilisateurs similaires. Hofmann et al [HB05] proposent ainsi une méthode à base de noyaux multiples combinés. Pour une application commerciale comme Amazon[LSY03] la communauté d'utilisateurs ainsi que le nombre d'objets traités peut être très grand tandis que les recommandations doivent être rapides et très ciblées pour générer des ventes. Ils tendent alors à faire leur propre système de recommandation.

Le domaine des recommandations utilise le terme d'apprentissage collaboratif mais principalement dans un sens d'imitation. Il s'agit de recommander ce qu'une personne avec des goûts similaires a également apprécié. Les goûts d'une personne sont évalués à partir de notes qu'il attribue à différents contenus. Ce domaine reste assez distant de l'apprentissage collectif en raison du manque d'interactions entre agents bien qu'une faible coopération soit présente, en partageant leurs notes, et qu'ils bénéficient ainsi des apports des autres. L'exploitation du contenu peut apporter un peu de profondeur en identifiant dans un profil plusieurs centres d'intérêt comme le fait Fab [BS97] et en enrichissant les recommandations d'un argumentaire sur les raisons de recommandation [PZ01]. On trouve également une collaboration complémentaire lorsque les utilisateurs enrichissent le contenu d'un objet par des tags [DZM⁺09] ou des commentaires.

2.2.5 Approches psychologiques et sociales

Les publications de Dillenbourg [DBBO95, Dil99b] témoignent du rapprochement possible entre la collaboration étudiée au niveau psychologique

et sociale et au niveau de l'intelligence artificielle distribuée. Il fait une présentation complète et instructive des travaux dans ce domaine dont seul une petite partie est reprise ici.

Deux grands courants de pensée concernent l'influence des interactions sociales sur l'apprentissage, le courant socio-constructiviste et le courant socio-culturel. Tout d'abord, Piaget [Pia23] s'intéresse essentiellement à l'aspect individuel du développement cognitif qu'il considère indissociable de l'action. Il met en avant deux moteurs à l'apprentissage, le conflit et la coordination des points de vue. Il voit l'interaction sociale comme un catalyseur pour l'apprentissage individuel. Le courant socio-constructiviste s'inspire des travaux de Piaget [DM84] pour se concentrer sur l'influence des interactions sociales sur le développement personnel. La thèse soutenue est que c'est en confrontant son point de vue avec les autres et par les interactions qui s'en suivent qu'un individu maîtrise de nouvelles approches.

Le courant socio-constructiviste a montré l'existence d'un bénéfice au niveau individuel de l'apprentissage collaboratif pendant et après la phase de collaboration [DM84]. Ils ont pour cela mené des évaluations individuelles avant et après une phase collaborative [RR88]. Cette phase collective peut être entre deux enfants de développement équivalent. Certaines des connaissances acquises ne sont alors pas une simple transmission d'un enfant plus avancé vers l'autre, mais bien issues de la confrontation des points de vue. La remarque qui permet de passer ensuite à un niveau socio-culturel est que ce n'est pas tant le désaccord en lui-même, mais la communication entre pairs qui s'en suit, qui permet de progresser [Bla88].

Le second grand courant, socio-culturel, prend son origine dans les travaux de Vygotsky. Il définit notamment le concept de "Zone Proximale de Développement (ZPD)" [VC78] qui représente la différence entre les capacités d'un individu seul et ses capacités en présence de l'aide d'un autre individu. Il s'appuie ensuite sur deux phénomènes complémentaires, l'internalisation, qui permet de nourrir la réflexion personnelle à partir des interactions sociales mémorisées, et l'appropriation qui permet de réévaluer le sens donné à ses propres actes à partir des réactions d'autrui.

Ainsi, pour qu'une collaboration soit fructueuse il est important d'avoir des différences entre les individus mais également que cette différence ne soit pas trop importante. D'après le courant socio-constructiviste, dans une interaction entre deux individus il y a bien plus à retirer que la somme de connaissances de l'un et l'autre et c'est par la confrontation des points de vue que se passe le progrès cognitif individuel. Le progrès personnel est donc favorisé par les interactions sociales générées par les différences entre indivi-

dus. La ZPD indique à l'inverse que le progrès d'un individu est limité à une certaine proximité de ce qu'il pourrait réaliser seul, empêchant des individus trop différents de bénéficier d'une collaboration.

La coopération est une fonction de la différence entre les individus mais aussi de la taille du groupe ainsi que d'un minimum de développement personnel. En effet des groupes de taille 3 tendent à être plus compétitifs et moins efficaces que des groupes de 2 [TD84] mais cette tendance disparaît lorsque les enfants sont libres d'interagir avec les autres au sein d'une classe [CL87]. Piaget détermine des stades de développement de l'enfant [Pia23] et certaines interactions ne sont tout simplement pas possibles tant qu'il ne les a pas atteints. De même des agents en intelligence artificielle [Gas91] ne disposant pas de la même sémantique sont inaptes à reconnaître qu'il existe un conflit entre leurs points de vue. De nombreux travaux soutiennent l'importance de la présence de différences entre les individus tant qu'elle ne devient pas trop importante [Kuh72, Web91].

Un autre travail lie les sciences sociales et l'apprentissage artificiel distribué, Jars [Jar05] insiste particulièrement sur l'importance du langage. Elle développe un modèle d'agent, ALONE, où un apprenti est spectateur d'un échange en langage naturel entre plusieurs agents experts bienveillants. Le langage est considéré comme très important et pas uniquement comme vecteur d'information mais aussi pour structurer la pensée et intégrer un contexte social.

Comme l'explique Dillenbourg [Dil99b] dans son chapitre introductif, l'apprentissage collaboratif a plusieurs significations et utilisations au sein de chaque domaine académique et d'autant plus dans des domaines distincts. Plutôt que de chercher un consensus, il est intéressant de chercher à réunir des points de vues différents et multidisciplinaires. L'apprentissage artificiel et en particulier l'apprentissage distribué se sont développés de manière indépendantes des théories cognitives et linguistiques. Le fossé entre les domaines est grand et les travaux qui les rapprochent, comme celui de Jars [Jar05], sont rares. En particulier, il est difficile de tenir compte d'un contexte social et affectif en apprentissage artificiel et plus courant de modéliser l'aspect cognitif.

2.2.6 Problématiques économiques en sciences régionales

Les travaux concernant l'apprentissage collectif dans ce domaine expliquent le succès d'ensembles d'entreprises fournissant des produits et services de haute technologies et localement fortement connectées connues sous le nom

de milieu innovateur. La notion d'apprentissage collectif en milieux innovateurs a été mise en avant par Camagni [Cam91]. Il explique que la connaissance est essentielle pour la réussite en milieu compétitif mais que dans un tel milieu elle n'est pas nécessairement transmise directement. Il pousse plutôt à s'intéresser au contexte social, à la présence d'un réseau fortement connecté entre les entreprises et aux mouvements de main d'œuvre en particulier. Ces travaux sont repris et approfondis avec l'ajout entre autre de nouvelles études de cas [LL99, Cap07, KW99]. Les principaux mécanismes d'apprentissage et transmission de connaissances qui sont identifiés sont les connexions fortes entre clients et fournisseurs, les collaborations formelles et informelles entre entreprises, les mouvements de personnels très qualifiés inter-entreprises et la création de nouvelles entreprises par des membres d'entreprises, universités ou autres partenaires. Cette forte connectivité, interne au milieu, peut favoriser ses membres tout en tenant à distance des compétiteurs extérieurs mais peut aussi conduire à un enfermement dans des stratégies peu compétitives et des connaissances obsolètes. Il est donc important pour un milieu innovateur de ne pas compter uniquement sur ses forts liens internes mais aussi d'apprendre de sources extérieures au milieu.

Larsson [LBHS98] traite de l'évolution d'une relation collaborative entre deux entreprises selon leur stratégie. Cette analyse va au delà du choix traditionnel entre collaboration et compétition. Il souligne l'importance d'étudier à la fois la balance entre transparence et réceptivité d'une entreprise, et celle entre intégration et distribution, le tout inscrit dans son contexte et son évolution temporelle. La dimension d'intégration correspond au profit total sur les deux entreprises tandis que la distribution correspond à la proportion du bénéfice personnel par rapport au bénéfice commun.

Ce domaine est très proche de la notion d'apprentissage collectif en raison du besoin de bénéfice et de la présence d'interactions complexes entre les entreprises. Dans un milieu économique, la collaboration entre entreprises ne peut se contenter d'être profitable à l'ensemble du groupe, il faut que chaque partenaire y soit gagnant individuellement pour que la collaboration perdure. Des préoccupations d'adversité viennent donc souvent compléter celles de collaboration. De plus les interactions et échanges entre entreprises ne sont pas du tout de même nature que entre deux agents humains ou informatiques.

2.2.7 Argumentation

Nous noterons tout d'abord qu'en science sociale, on reconnaît l'importance de fournir une explication plutôt qu'un simple résultat. Webb [Web91]

montre que le fait de fournir une explication a un impact positif sur la personne qui la formule tandis que l'absence d'explication peut avoir un impact négatif sur quelqu'un souhaitant en recevoir. Il montre également que dans des groupes d'élèves ces explications sont naturellement plus fréquentes lorsqu'ils sont de niveau modérément différents. Des élèves moyens en groupe avec des élèves de niveau plus faible ou plus élevés bénéficient ainsi plus de l'apprentissage collaboratif que des groupes de niveau uniformément élevés ou bas qui ne veulent ou ne peuvent pas fournir d'explication.

Une approche complémentaire intéressante [Bet12] considère l'influence de diverses techniques d'argumentation lors d'une recherche de consensus. Chaque argument est ici constitué de prémisses et d'une conclusion. Les attitudes adoptées peuvent être orientées vers soi ou vers la personne à convaincre, selon que les arguments sont construits sur les prémisses personnelles ou celles avancées par l'opposant. L'argumentation est de plus considérée agressive si elle vise à critiquer le point de vue adverse plutôt qu'à soutenir le sien. Selon ces critères et la présence de connaissances communes a priori, les échanges sont plus ou moins enrichissants pour chacune des parties et le consensus est plus ou moins orienté vers l'un ou l'autre voir impossible à atteindre. Ces études indiquent comme préférable de se baser sur les connaissances adverses pour fournir de nouveaux arguments et d'être d'autant plus agressif que les points de vue sont lointains.

L'argumentation en fouille de données peut être utilisé en phase de restitution pour déterminer un résultat sous la forme d'une recherche de consensus [WCBC12] ou pour former une nouvelle théorie centrale indépendante [XYL15] ou encore pour nourrir un apprentissage individuel [OP14, BES07].

Wardeh et al [WCBC12] définissent PISA puis MABCE comme processus de classification basé sur l'argumentation entre plusieurs agents. Un agent par classe envisagée dispose de son propre ensemble de données personnelles et défend, tel un avocat, la possibilité pour chaque exemple d'entrer dans la classe qu'il lui attribue. Les arguments sont toujours un agrégat des données afin d'en préserver la nature privée. Il considère comme important d'utiliser des données limitées locales plutôt que globales pour donner de l'importance à des phénomènes locaux facilement ignorés lorsqu'ils deviennent mineurs à une échelle globale. Un exemple avancé est l'attribution d'aide financière pour la santé dans une région minière ou le poumon est souvent atteint. Un autre exemple est la notation de projets ou dissertations dans une formation académique lorsque l'évaluation est effectuée par deux personnes distinctes.

Avec AMAJL, Xu et al [XYL15] mettent l'accent sur l'utilisation de l'argumentation dans le cadre d'apprentissage d'ensemble. Des agents séparés

disposant de données distinctes apprennent individuellement une base de connaissance, sous la forme d'un ensemble de règles. Ces agents doivent ensuite argumenter sur différents sujets qui leur sont soumis pour qu'un agent central puisse en extraire les règles de meilleure qualité pour constituer la connaissance globale.

Les travaux d'Ontañon et Plaza avec A-MAIL [OP14] utilisent en priorité des arguments agrégés par souci de conserver les données privées mais considère également l'utilisation de données brutes comme argument. SMILE proposé par Bourgne et al [BES07] n'utilise que les données brutes, mais comme pour A-MAIL le partage des exemples est parcimonieux. Les exemples fournis dépendent de l'état actuel de l'apprentissage des autres agents.

L'argumentation est un processus de coopération avancé. C'est une méthode d'interaction particulièrement intéressante lorsque l'on considère des agents autonomes et complexes pouvant représenter des humains. L'argumentation devient un outil de l'apprentissage collectif lorsque l'on considère son impact à un niveau individuel comme dans le cas de A-MAIL ou de SMILE.

2.2.8 Apprentissage collectif

Les objectifs de l'apprentissage collectif varient en fonction du nombre et de la complexité des agents. On distingue en particulier deux grands domaines. Des agents simplistes regroupés en essaims, parfois très nombreux, sont étudiés principalement au niveau du groupe. Des agents cognitifs, par nature plus complexes, peuvent être étudiés plus naturellement au niveau de l'individu bien que les résultats au niveau du groupe soient également intéressants.

Essaims

Une sous-partie de l'apprentissage collectif concerne la décision collective dans le contexte de troupeaux ou d'essaims (swarm-intelligence). L'inspiration vient de l'étude des fourmis [BDGP90, MPF01], des abeilles [SCS91, SVS⁺12] et d'autres animaux grégaires [SFKD99] pour en arriver à des études et méthodes d'apprentissage applicables sur des robots [COH09, GCMH⁺10, VFHD16]. La prise de décision est alors totalement décentralisée, réparties sur plusieurs individus qui n'ont accès qu'à des informations locales et communiquent avec leurs semblables les plus proches. Des décisions qui peuvent paraître complexes si elles doivent être réalisées par un seul individu relèvent en réalité au niveau collectif de capacités très basiques et communications

simple par les membres du groupe. Par exemple les abeilles déterminent ainsi la source de nectar la plus profitable [SCS91] tout simplement par le nombre de butineuses qui y vont régulièrement. Dans ce contexte l'interaction entre les individus est forte du fait de leur proximité spatiale. Les robots à la recherche d'algues toxiques [COH09] doivent tenir compte de la présence des autres pour ne pas entrer en collision, obtenir des relevés suffisamment lointains pour être utiles et rester suffisamment proches pour maintenir la communication avec le groupe. Ils sont autonomes puisqu'ils doivent prendre individuellement la décision de leur prochain déplacement et bénéficient tous des relevés de leur voisins pour aboutir à cette décision. Le fonctionnement de décision collective d'un essaim peut s'apparenter à une représentation neuronale du cerveau. Seeley et al [SVS⁺12] étudient ainsi l'importance de signaux d'arrêt de processus de décision collective pour éviter les inter-blocages en prenant pour exemple des abeilles éclaireurs.

L'intelligence d'essaim est caractérisé par une très grande mobilité des individus qui la compose. On observe également dans ce cas principalement le bénéfice au niveau du groupe bien que tous les individus en profitent.

Agents cognitifs

On souhaite s'intéresser à des phénomènes d'apprentissage collectif dans des réseaux moins mobiles composés d'agents plus complexes. Chaque agent est autonome de telle sorte qu'il puisse apprendre et restituer seul, mais également bénéficier de la communication et être actif au sein d'un groupe. (figure 2.6e) C'est un apprentissage collaboratif, a priori bienveillant. Il s'agit d'apprendre ensemble à savoir faire seul. De tels réseaux peuvent être des réseaux pair-à-pair ou des réseaux sociaux tels que des chercheurs d'un même domaine reliés par leurs co-publications et rencontres lors de projets ou conférences.

Bien que l'apprentissage artificiel collectif soit rarement étudié, au moins deux approches répondent à cette forme d'apprentissage, A-MAIL (Argumentation-based Multi-Agent Inductive Learning) présenté par Ontañón et Plaza [OP10, OP14] et SMILE (Sound Multi-agent Incremental LEarning) présenté par Bourgne et al [BES07].

L'objectif commun pour ces deux approches est que chaque agent forme une hypothèse qui soit consistante avec l'ensemble des données détenues par l'ensemble des agents sans avoir à partager l'intégralité de ces données. Les agents obtiennent de nouveaux exemples incrémentalement et les conservent en mémoire pour les réutiliser ultérieurement. Ces deux algorithmes d'ap-

prentissage sont conçus autour des mêmes succession de phases : (1) révision par un agent de son hypothèse par rapport à sa mémoire d'exemples, (2) échange argumentatif avec les autres agents, (3) révision de son hypothèse pour tenir compte des arguments des autres agents.

A-MAIL diffère de SMILE en plusieurs points. Tout d'abord, il envisage des arguments de deux natures, des exemples et des règles tandis que SMILE se limite à des contre-exemples. Ensuite, A-MAIL recherche une haute précision plutôt qu'une consistance exacte. Il accepte une règle dès lors qu'elle atteint un certain seuil d'acceptabilité τ représentant le nombre d'exemples de la bonne classe couverts par la règle sur le nombre d'exemples totaux couverts. De plus, le protocole ne cherche à rectifier au niveau multi-agents que les règles incorrectes sans tenir compte de la complétude de l'ensemble des règles formant l'hypothèse. La complétude d'une hypothèse n'est assurée que vis-à-vis de la mémoire personnelle de l'agent et la valeur de rappel est donc plus faible. SMILE considère la consistance d'une hypothèse dans son ensemble comme présenté dans la définition 1. Enfin, A-MAIL effectue des phases argumentatives avec l'ensemble des agents présents en même temps, sans contrainte de réseau et se limite à un nombre d'agents plus faible (2 à 20). Les théories individuelles sont envoyées à l'ensemble des autres agents puis l'argumentation s'effectue en tour par tour, au moyen d'un jeton, circulant entre les agents, qui leur donne la parole. Les agents peuvent alors attaquer une règle constitutive de la théorie d'un autre agent à l'aide d'un argument formé d'une règle τ -acceptable moins générale ou d'un contre-exemple. Les agents peuvent également défendre leur règle en attaquant, à leur tour, de la même manière, l'argument qui la met en défaut. Deux formats d'agent sont possibles. Si l'agent est empiriste il vérifie la τ -acceptabilité de chaque argument par rapport à sa propre mémoire d'exemples et considère comme rejeté, un argument qui n'atteint pas ce seuil. Si l'agent est à l'inverse rationaliste, il accepte tous les arguments actuels les plus spécifiques et en déduit la valeur accepté ou rejeté de tout autre argument précédent en fonction de sa position dans l'échange argumentatif.

Bien que l'idée de limiter les exemples transmis en utilisant des règles comme arguments soit intéressante, elle n'est pas applicable lorsqu'on recherche également une grande précision. Dès que l'on attend des hypothèses et règles formulées qu'elles classent correctement une haute proportion des exemples, c'est à dire que le seuil d'acceptabilité τ est élevé, le nombre de règles réellement échangées décroît au profit des exemples.

La dimension collective insiste sur l'intérêt personnel et l'individualité des membres constitutifs d'un groupe.

La décision collective en essaim montre le comportement d'un groupe

d'individus simplistes, similaire à celui que pourrait avoir un individu unique disposant de facultés et connaissances beaucoup plus complexes. Il montre l'émergence d'un savoir collectif basé sur des interactions minimalistes.

L'apprentissage collectif d'agents cognitifs considère que la connaissance collective est le niveau de connaissance recherché par chaque individu qui le constitue. Les individus constitutifs de ce genre de groupe sont des individus complexes. Ils ne nécessitent pas la présence d'autres individus pour apprendre mais peuvent en bénéficier au moyen d'interactions collaboratives.

Nous prendrons le parti de nous baser sur SMILE pour ensuite l'étendre à une plus grande dynamique qui englobe à la fois les communications simultanées d'un agent avec plusieurs partenaires malgré des contraintes de réseau, l'arrivée multiple d'exemples ainsi que l'apprentissage et les communications de plusieurs agents en parallèle.

2.3 Réseau

2.3.1 Les structures de réseaux

Nous considérons des systèmes multi-agents tels que les agents ne partagent pas nécessairement un même environnement mais sont reliés entre eux par des moyens de communication. On peut représenter ce réseau de communication par un graphe dont la structure peut influencer le résultat de la collaboration du SMA.

Différentes propriétés de graphes

Les graphes étudiés représentent le réseau de communication d'un SMA. Les agents sont les sommets du graphe et les arêtes représentent la possibilité d'envoyer un message. On considère que toute communication est bilatérale, donc aucune arête n'est orientée et elles ont toutes le même cout d'utilisation. Les graphes étudiés sont majoritairement des graphes **connexes** c'est-à-dire qu'il existe un chemin constitué d'arêtes entre toutes paires de sommets.

Pour décrire ces graphes on parlera beaucoup de **distance** entre deux agents A et B , il s'agit du plus petit nombre d'arêtes à parcourir pour aller de A à B . Le **diamètre** d'un graphe sera alors la plus grande distance entre deux sommets de ce graphe. La **distance moyenne** d'un graphe est la moyenne des distances entre toutes paires de sommets de ce graphe.

La **densité** représente le rapport du nombre d'arêtes présentes sur le nombre d'arêtes possibles et varie de 0 pour un graphe sans arêtes à 1 pour une clique.

Le **degré** d'un sommet est le nombre d'arêtes qui en partent et le **degré moyen** d'un graphe la moyenne des degrés de chacun des sommets du graphe. Le degré moyen $Degm$ d'un graphe est directement lié à la densité D d'un graphe et au nombre total N_s de sommets par la formule : $D * (N_s - 1) = Degm$.

Les termes définis dans ce paragraphe sont illustrés par l'exemple 2.

Exemple 2

- 4 sommets (agents) et
4 arêtes (communications)
- distance ($Ag1, Ag3$) : 1 ;
distance ($Ag2, Ag3$) : 2
- diamètre : 2
- distance moyenne : $(4*1+2*2)/6 = 4/3$
- nombre de chemins minimaux de longueur 2 : 2
- densité : $4/6 = 2/3$
- degré moyen : $(3+2+2+1)/4 = 2$

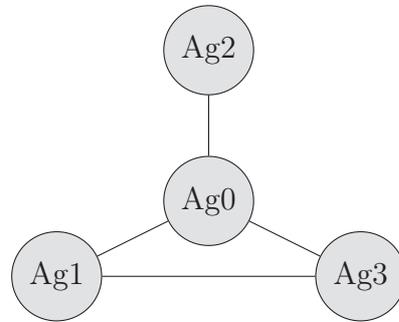
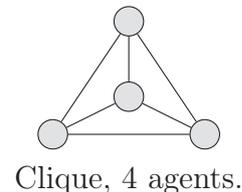


FIGURE 2.7 – Quelques données de graphe illustrées sur un petit graphe.

Différentes structures communes

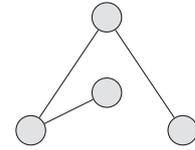
Différentes structures de graphes pourront être utilisées :

- Clique
 - Définition : graphe complet, ayant toutes les arêtes possibles.
 - Utilisation : peut représenter un groupe de travail non hiérarchisé sans aucune limitation des communications comme lorsque les individus qui le composent sont réunis dans un même lieu.
 - Génération : définir le nombre de sommets n puis ajouter une arête entre chaque paire.



- Ligne

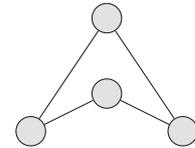
- Définition : graphe connexe, sans cycle, tel que chaque sommet n'a qu'un ou deux voisins.
- Utilisation : cas de relation extrêmement hiérarchisée ou encore une connexion filaire longue distance.
- Génération : ajouter les sommets un à un avec une arête le reliant au précédent.



Ligne, 4 agents.

- Cercle

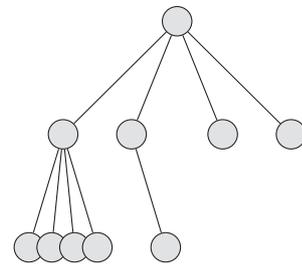
- Définition : graphe connexe, sans cycle, tel que chaque sommet a exactement deux voisins.
- Utilisation : cas de circuit fermé.
- Génération : ajouter les sommets un à un avec une arête le reliant au précédent, le dernier étant de plus relié au premier.



Cercle, 4 agents.

- Arbre Arb_k

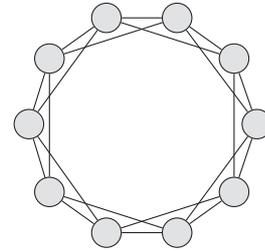
- Définition : graphe d'arbre régulier à k fils.
- Utilisation : forme de structure hiérarchique.
- Génération : en partant d'un premier sommet, pour chaque sommet, ajouter le maximum de sommets fils entre k et le nombre de sommets restant. Ajouter également une arête entre chaque sommet père et sommet fils.



Arb4, 10 agents.

- Régulier Reg_k

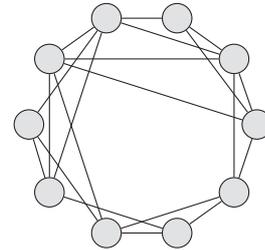
- Définition : graphe régulier connexe circulaire tel que chaque sommet est connecté à ses k plus proches voisins.
- Utilisation : comme un cercle mais où la proximité permet d'ajouter des redondances de communication.
- Génération : les sommets disposés en cercle sont reliés à leurs $k/2$ précédents et $k/2$ suivants.



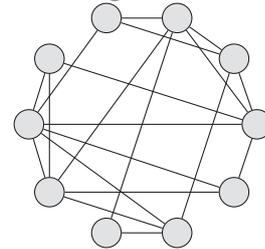
Reg4, 10 agents.

- Small-World SmW_kpp

- Définition : ils sont appelés ainsi pour signaler que tout sommet peut atteindre tout autre sommet en parcourant peu d'arêtes.
- Utilisation : nombreux cas de réseaux auto-organisés (sociaux, pair-à-pair, ...).
- Génération : algorithme de Watts-Strogatz [Wat98] : les graphes sont construits sur la base du reg_k , le treillis régulier dont les sommets disposés en cercle sont reliés à leurs n plus proches voisins situés de part et d'autre. Chaque arête $((n_i, n_j) | i < j)$ a une certaine probabilité p de modifier n_j en un autre sommet n_k selon une loi uniforme sur l'ensemble des sommets hors n_i et n_j .



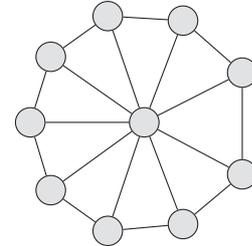
SmW4p01, 10 agents.



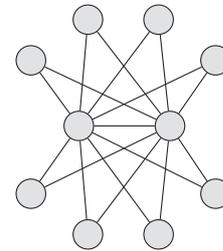
SmW4p05, 10 agents.

- MultiPôles MultiP*kdd*

- Définition : les Multipôles sont constitués de deux types de nœuds, les sommets principaux (pôles) qui sont reliés en priorité à un maximum d'autres nœuds et les sommets secondaires. On considère trois types d'arêtes selon les types de sommets reliés.
- Utilisation : relations centralisées autour de personnes ou serveurs de référence.
- Génération : pour un graphe MultiP*kdd*, *k* est le nombre de pôles et la quantité d'arêtes à placer est déterminée par la densité *d*. Ils sont créés en reliant en premier les pôles entre eux jusqu'à former une clique de pôles puis les pôles aux sommets secondaires en équilibrant la répartition sur les pôles et sur les sommets secondaires et enfin les arêtes entres sommets secondaires.
Une version dite séparée ignore la première étape et ne relie aucun des pôles entre eux.



Roue, 10 agents :
MultiP1d04.



Bipôle, 10
agents :
MultiP2d04.

Structures préférentielles en apprentissage collectif

L'apprentissage collectif s'apparente soit à des groupes d'individus dans un même lieu soit à des individus distants appartenant à un réseau auto-généré tel qu'un réseau pair-à-pair ou un réseau social. Dans le cas d'un lieu commun, il n'y a pas de limitation de communication et on peut modéliser la communauté par une clique. Dans le cas d'un réseau auto-organisé, la structure la plus commune se trouve entre une structure régulière et une structure aléatoire sous la forme de réseaux petit-monde (Small-World).

Watts et Strogatz [Wat98] ont particulièrement étudié cette structure par comparaison avec un graphe régulier. Selon leur algorithme on peut former de tel réseaux à partir d'un graphe régulier et d'une certaine probabilité *p* de modification de chaque arête. Il suffit d'une valeur de *p* très faible pour diminuer grandement la distance moyenne d'un réseau tandis que ce phénomène n'est pas visible localement, au moyen d'un coefficient de clustering local (nombre de connexions entre voisins d'un individu divisé par nombre de connexions possibles), avant de grandes valeurs de *p*. L'effet petit monde est atteint avec un nombre très faible de raccourcis depuis une structure régu-

lière. Il est alarmant lorsque l'on considère une épidémie et rassurant lorsque l'on considère un apprentissage basé sur un mécanisme local.

Bien que la structure de Small-world ne soit pas la seule à envisager, sa fréquence en fait une structure particulièrement importante à étudier.

2.3.2 Influence de la structure de réseau

Influence sur les résultats individuels et collectifs

De nombreux travaux étudient l'influence de la forme d'un réseau sur les résultats produits par un SMA. Nous détaillerons deux travaux dans le domaine collaboratif.

Mason et Watts [MW12] effectuent une expérience avec des agents humains dans le cadre d'un jeu appelé "Wildcat Wells". Il s'agit de trouver dans un désert des puits de pétrole cachés. Le jeu se déroule en plusieurs tours. Chaque participant peut visiter un lieu de la carte à chaque tour et en retire un score en fonction de la richesse et la proximité de puits alentours. L'expérience est rendue collaborative en reliant les participants par un réseau tel que chaque participant peut consulter ses localisations et scores des tours précédents mais aussi ceux de ses voisins. Les participants sont payés en fonction de leur score personnel total et non de leur score relatif. Ils sont donc motivés par maximiser leur gain sans nécessité d'induire les autres en erreur. Ces expériences sont menées en appliquant entre les joueurs des réseaux de caractéristiques variées mais conservant tous leurs sommets de degré 3. Ils expliquent les bons scores par un bon équilibre entre exploitation des résultats actuels et exploration de l'espace.

De nombreux résultats intéressants soulignent l'influence du réseau sur la découverte de l'optimum de la carte et sur le score obtenu. La remarque la plus immédiate est que les personnes reliées par un réseau ont de meilleurs scores qu'un groupe témoin où chaque participant est isolé. Ils évaluent ensuite l'efficacité d'un réseau à sa distance moyenne courte et montrent que les scores s'améliorent avec l'efficacité du réseau. Ils reprennent l'idée du compromis entre exploitation et exploration sous forme d'un dilemme social. L'exploitation du meilleur résultat actuel garantit un meilleur score sur une courte durée, mais ne garantit pas de trouver l'optimum de la carte, à l'inverse de l'exploration. Ainsi un participant qui copie beaucoup tend à améliorer son score individuel mais pénalise le score collectif au risque si les autres font comme lui de pénaliser également son propre score. C'est un dilemme similaire aux situations étudiée par l'apprentissage collectif en milieu innovateur [Cam91]. La structure du réseau a donc une influence sur le résultat moyen

que peut espérer atteindre un participant mais la position occupée dans le réseau a également de l'importance au niveau individuel. Les individus centraux sont favorisés. Un réseau moins centralisé favorise à l'inverse la réussite collective. Selon qu'un participant préfère un gain relatif ou absolu le réseau qui lui convient n'est pas le même.

Dans une expérience de décision collective, à partir de robots minimalistes [VFHD16], Valentini montre un compromis entre la vitesse à laquelle l'essaim converge vers une solution et la justesse de la décision prise. Il peut décaler ce compromis en modifiant le nombre de voisins dont chaque agent tient compte lors de décisions individuelles.

Influence sur l'apprentissage collectif

Zollman [Zol07, Zol13] présente de nombreux phénomènes concernant le savoir collectif, sa propagation et modification essentiellement, plus que sa genèse. Il considère qu'il est possible de faire des communications sans perte et utilise des modèles simplistes pour représenter les connaissances. Il considère la question des structures de graphe selon de nombreux critères comme une répartition des coûts de communication équitable ou la limitation des communications entre agents suffisamment similaires voire à l'inverse suffisamment différents. Il pose le problème de la possibilité d'atteindre un état stable ou un consensus pour un réseau d'agents ayant initialement des modèles divers. Bien que son étude se limite à communiquer un modèle sans argumentation, elle s'étend en partie à l'apprentissage actif et couvre de nombreux cas. Il développe en particulier le problème selon lequel un agent cherche à déterminer parmi plusieurs experts celui qui a le meilleur rendement tout en maximisant ses chances de gains. Ces notions peuvent être représentées par des fonctions de probabilité à partir des statistiques de ses essais déjà réalisés. Dans ce cas particulier un agent au sein d'une structure présentant plus d'arêtes et donc de communications est moins sûre de converger vers la bonne solution que s'il se trouve dans une structure minimaliste tel qu'un cercle. Ce phénomène peut être observé dans la société comme pour l'exemple médical de la découverte de l'origine bactérienne de l'ulcère gastro-duodénal (peptic ulcer disease) qui a valu un prix Nobel à Warren et Marshall. Cette découverte a longtemps été retardée par la présence d'une personne influente, très écoutée dans son domaine qui soutenait de manière convaincante, une thèse opposée.

Dans ce document, Bien que l'on ne modélise pas les échanges d'informations à un même niveau, d'une part parce que l'on modélise l'apprentissage par induction plutôt que par probabilité, et d'autre part parce que

l'apprentissage considéré est non bruité, on retiendra plusieurs principes. Il est intéressant de tenir compte de l'influence de la structure de réseau sur l'apprentissage qui, selon ses caractéristiques, peut accélérer ou limiter un apprentissage.

On reconnaîtra également un effet bénéfique de la remise en question régulière de parties de la théorie obtenue, tout en profitant d'une base de savoir déjà construite par un apprentissage incrémental.

2.3.3 Les dynamiques de réseau

Le dynamisme représente la variation au cours du temps. Nous distinguerons trois formes de dynamiques dans un réseau. Au cours du temps, les variations affectent la structure du réseau et l'information qu'il contient. L'information est doublement dynamique, sa répartition change au moyen de la diffusion mais sa nature change également suite à des révisions. Le dynamisme du réseau peut, par ailleurs, être considéré comme plus ou moins important en fonction de la fréquence de ses variations et de leur simultanéité plutôt que succession.

Nous verrons dans cette partie la variation de la structure du réseau puis la diffusion de l'information et en particulier les protocoles d'agrégation de données. La variation de l'information par révision ne sera pas beaucoup développée dans cette partie mais est déjà couverte par l'apprentissage incrémental précédemment présenté. La réalisation des modifications en parallèle sera par contre une thématique importante dans cette thèse.

Structure variable

Un apprentissage dynamique fait souvent référence à la variation de la structure qui lie les agents. C'est une problématique très étudiée dans les réseaux pair-à-pair [WBK09] où il est toujours possible qu'un agent se connecte ou se déconnecte en tout point du réseau. Il peut également s'agir d'agents mobiles liés par une proximité physique, ce qui est plus souvent le cas avec des robots. Des problématiques de pannes [VFHD16] peuvent survenir ou bien comme dans le cas d'essaims on peut chercher à contraindre les agents dans l'espace pour les forcer à conserver une connectivité [VFHD16, COH09].

Le protocole SMILE a été confronté à cette problématique de connexions et déconnexions [BES09] mais le sujet ne sera pas développé dans cette thèse. Il reste cependant compatible avec les protocoles proposés ici et sera évoqué dans la partie discussion.

Diffusion d'information

Deux grands domaines sont préoccupés par les phénomènes de diffusion dans un réseau, l'étude de la rumeur et l'épidémiologie. Selon l'objectif, on peut chercher à modéliser une propagation vers l'ensemble des autres agents, une partie d'entre eux ou un individu spécifique.

Un problème classique de la distribution d'information est le problème des commères : chaque agent connaissant au départ une information distincte (appelée rumeur), le but est que chaque agent connaisse toutes les rumeurs [EM89]. Une méthode gourmande en réseau pour qu'une information atteigne tous les agents se présente telle que chaque agent qui reçoit l'information l'envoie une fois à tous ses voisins. Cette technique, appelée routage par saturation ou flood, est robuste mais couteuse en communications. Elle ne nécessite aucune connaissance à priori du réseau.

Une autre méthode plus légère mais moins garantie, consiste à choisir successivement et de manière aléatoire plusieurs voisins pour leur transmettre une rumeur [KSSV00]. Leur nombre est choisi de telle sorte qu'il y ai une haute probabilité d'avoir atteint tous les autres agents. Ce sont des algorithmes dit épidémiques.

Concernant la transmission d'information d'un point à un autre dans un environnement mobile, on peut trouver des solutions du coté des protocoles de routage [JM96].

En apprentissage collectif exact, le problème est différent puisqu'il faut contacter tous les autres agents plutôt que de trouver un chemin efficace vers un seul d'entre eux comme pour les modèles de rumeurs. À l'inverse d'un modèle traditionnel de rumeur cependant, l'information est susceptible d'être modifiée au fur et à mesure qu'elle est propagée dans un contexte d'apprentissage.

Agrégation d'informations

Un domaine entier s'intéresse à la diffusion d'une information dynamique qui doit être modifiée au fur et à mesure de sa propagation et du temps qui passe. Les protocoles d'agrégation de données sont particulièrement indiqués pour effectuer la transmission et la synthèse de l'information.

Les protocoles d'agrégation sont proches de l'apprentissage multi-agents. Il s'agit de déterminer l'état actuel général qui représente le plus justement l'ensemble des relevés de cas particuliers effectués par les différents nœuds. Jin Cui rappelle, dans sa thèse "Data Aggregation in Wireless Sensor Networks"

[Cui16], de nombreuses propriétés sur les protocoles d'agrégation de données, qui permettent de mettre en œuvre des réseaux de capteurs sans fil.

Les protocoles d'agrégation sont par nature hiérarchisés. Un seul agent, éventuellement mobile, a des requêtes à formuler et donc cherche à obtenir toute l'information. C'est une forme d'apprentissage centralisé. La plupart des nœuds se contentent de collecter et transmettre l'information tandis que certains intermédiaires doivent également condenser l'information pour limiter le flux.

Les problèmes auxquels sont confrontés les réseaux de capteurs sont des problèmes de quantité d'information qui sont très redondantes et fréquentes. L'information contenue dans l'ensemble des relevés des capteurs ne doit pas être connue par chaque capteur mais transmise à un point de collecte. Les problèmes à surmonter sont alors la taille limite de la bande passante, qui peut être saturée par la taille ou le nombre des messages transmis, et l'énergie consommée par un capteur [KEW02], lorsqu'il doit pré-traiter les informations qu'il possède ou réaliser une fonction d'agrégation complexe ou de nombreuses fois. Les données étant en effet fortement corrélées dans le temps et l'espace, il est possible de les synthétiser au fur et à mesure qu'elles se rapprochent de l'agent apprenti central. Les fonctions d'agrégation envisagées sont donc souvent très simples, pour limiter leur coût en énergie et en temps. On trouve parmi les plus courantes MAX - MIN - Moyenne - Somme ... Les fonctions d'agrégation peuvent être utilisées pour compresser les données [KL05] ou pour prédire une valeurs correspondant aux instants suivants [LWT05].

D'autres problématiques complémentaires se posent comme la sécurité [KD15] et la protection des données privées [ETpLPG13], la justesse de l'information finale agrégée par rapport aux données brutes ou encore la latence [LWHL14], c'est-à-dire le temps mis par l'information pour aller de l'acquisition par un capteur à son exploitation par l'apprenti.

Les structures de réseau envisagées doivent permettre à un destinataire final d'avoir accès à l'information récupérée par de nombreuses sources et sont donc essentiellement hiérarchisée. On trouve notamment des structures d'arbres [MFHH02] ou des épines dorsales [SKPP05] ou encore des grappes [HCB02]. Parfois, pour des soucis de répartition de la consommation d'énergie, la structure est régulièrement modifiée. Certains protocoles, pouvant se permettre une plus grande latence, vont jusqu'à se passer de structure [FLS07].

Certaines contributions au domaine de l'agrégation de données modifient l'ordre des priorités. L'article de Krishna P.N. Puttaswamy et al présentant Anonygator [PBP10] accorde une priorité à des considérations autres que l'économie d'énergie. Cet algorithme d'agrégation considère chaque nœud

comme un utilisateur montrant un intérêt important quant à son anonymat. La solution envisagée implique toujours la transmission de l'ensemble de leurs informations, mais sans qu'aucun intermédiaire, ni même le destinataire final, n'en connaisse la source. Il utilise pour cela des méthodes d'encapsulation et de parcours aléatoires jusqu'à la structure hiérarchisée d'agrégation. Des préoccupations supplémentaires de limitation de pollution sont alors à considérer.

L'apprentissage multi-agents considéré dans cette thèse ne présente pas autant de corrélation entre les données reçues par les différents agents. Les préoccupations liées à la redondance auront donc une importance bien moindre. On pourra considérer notamment que les agents souhaitent limiter les informations qu'ils communiquent à ce qui est strictement nécessaire à l'apprentissage. La latence par contre sera associée au temps d'exécution qui peut devenir nettement plus conséquent en raison de processus d'apprentissage plus complexe que des fonctions d'agrégation. L'apprentissage ne se restreint pas à la prédiction de l'instant suivant, qui a une grande proximité avec l'état actuel. Grâce à la formulation de règles générales, l'apprentissage permet de prédire des cas divers dans des conditions différentes. Enfin, la principale caractéristique de l'apprentissage collectif, qui diffère des protocoles d'agrégation, est que tous les agents sont apprentis. L'information ne peut être simplement centralisée, elle doit être accessible pour tous.

Conclusions sur les préoccupations dynamiques

Les protocoles d'apprentissage en réseau choisissent souvent d'appliquer un arbre sur le réseau pour en retirer tous les cycles potentiels. On souhaite effectivement que l'information se propage selon un arbre pour qu'elle atteigne une et une seule fois chaque agent sans générer de traitement redondant de l'information et garantir la terminaison du protocole. On peut souhaiter également que la construction de cet arbre soit dynamique et dépende de la réactivité des nœuds voisins à l'instant de sa propagation plutôt qu'un choix statique à priori. Bien que ce soit une problématique de routage, peu explorée en apprentissage, le protocole Waves proposé ici présente cette spécificité.

Une autre problématique liée au dynamisme concerne le rythme des variations du réseau. La volonté de permettre des modifications simultanées et de plus grande ampleur conduit à rechercher un fonctionnement en parallèle plutôt que séquentiel.

Les principales caractéristiques dynamiques étudiées ici concernent donc

la réactivité du réseau lors de la diffusion de l'information et la coexistence possible de plusieurs mécanismes de révision et de diffusion. La dimension dynamique de l'apprentissage a donc beaucoup motivé le travail de recherche de cette thèse bien qu'elle soit assez différente des problématiques dynamiques classiques.

Chapitre 3

SMILE

3.1 Protocole pour des agents tous connectés

3.1.1 Communication selon le protocole SMILE

Dans le cadre de l'apprentissage de concepts sans erreur, tenir compte des connaissances du groupe revient à avoir une hypothèse consistante vis à vis des exemples reçus par l'ensemble des agents du système. Une telle hypothèse sera dite *SMA-consistante*. On souhaite que chaque agent ait une hypothèse SMA-consistante et pour cela il doit la confronter à ses exemples, mais aussi aux exemples de chacun des autres agents du système. La consistance d'une hypothèse par rapport à un ensemble d'exemples est *compositionnelle*, signifiant qu'une hypothèse est consistante avec l'union de deux ensembles d'exemples si et seulement si elle est consistante avec chacun d'eux. Ainsi pour obtenir une hypothèse SMA-consistante, un agent peut se contenter de vérifier qu'elle est consistante avec l'ensemble des exemples de chaque agent indépendamment. Le protocole SMILE (Sound Multi-agent Incremental LEarning) définissant les interactions entre agents par envoi de messages est basé sur ce principe. Ce protocole est introduit dans un article de Bourgne et al [BES07].

Un agent SMILE dispose d'une mémoire d'exemples permettant de retenir à la fois les exemples qu'il collecte lui-même depuis son environnement, appelés *exemples internes*, et ceux qui lui sont communiqués par d'autres agents, appelés *exemples externes*. Il dispose d'une unique hypothèse personnelle qui lui permet de répondre à une phase de restitution et qu'il peut mettre à jour à la réception de nouvelles informations.

Le protocole SMILE se définit **au niveau du réseau** comme une succession de *révisions globales* :

Lorsqu'un agent A_a est amené à réviser son hypothèse, souvent pour tenir compte d'un nouvel exemple, il tire parti de la communauté pour s'assurer qu'elle est SMA-consistante. Pour cela A_a propose son hypothèse H_p tour à tour à chacun des autres agents A_c . Une réponse négative d'un agent A_c permet à A_a d'affiner son hypothèse H_p par une nouvelle révision en H'_p , avant de la proposer de nouveau, tandis que les réponses positives de l'ensemble des agents lui confirment la SMA-consistance de son hypothèse révisée. Pendant cet échange appelé *révision globale* on dit que A_a est apprenti et les autres (A_c) sont les critiques. Les agents critiques n'ont alors pas modifié leur hypothèse depuis la précédente révision globale et celle-ci n'est peut être plus SMA-consistante. Un mécanisme d'*adoption* leur permet de remplacer leur hypothèse par H_p lorsqu'elle est reconnue SMA-consistante. Faire adopter H_p par les agents A_c permet de s'assurer que tous les agents ont une hypothèse SMA-consistante à la fin de chaque révision globale.

Le protocole SMILE (figure 3.1) peut également se définir **au niveau d'un agent** par un ensemble de comportements à adopter selon le message reçu :

- **new-example** ($newE$) : si l'hypothèse H_a de l'agent est consistante avec le nouvel exemple reçu $newE$ il n'y a pas besoin de révision ni au niveau de l'agent ni au niveau du SMA. Par contre si H_a devient inconsistante, l'agent commence une révision globale par la révision de H_a en H_p puis son envoi dans un message **propose** à son premier voisin. Il initialise un compteur de réponses positives à 0.
- **propose** (H_p) : en tant que critique, l'agent qui reçoit un message propose vérifie la consistance de l'hypothèse reçue H_p avec l'ensemble de ses exemples. Si au moins un de ses exemples est inconsistant avec H_p , il l'envoie en réponse dans un message **counter-example** (figure 3.2) sinon il répond par un **accept**.
- **accept** : l'agent apprenti qui a proposé H_p et reçoit un **accept**, augmente son compteur de réponses positives de 1. Si le compteur est égale au nombre de voisins, la révision globale se termine (figure 3.3) par la notification de la SMA-consistance de H_p à tous les autres agents dans un message **SMA-cons**. Sinon il propose son hypothèse au voisin suivant.
- **counter-example** (CE) : l'agent apprenti qui a proposé H_p et reçoit un **counter-example** mémorise le contre-exemple reçu CE puis révisé son hypothèse H_p en H'_p . Il réinitialise son compteur de réponses positives et propose H'_p à un premier voisin.

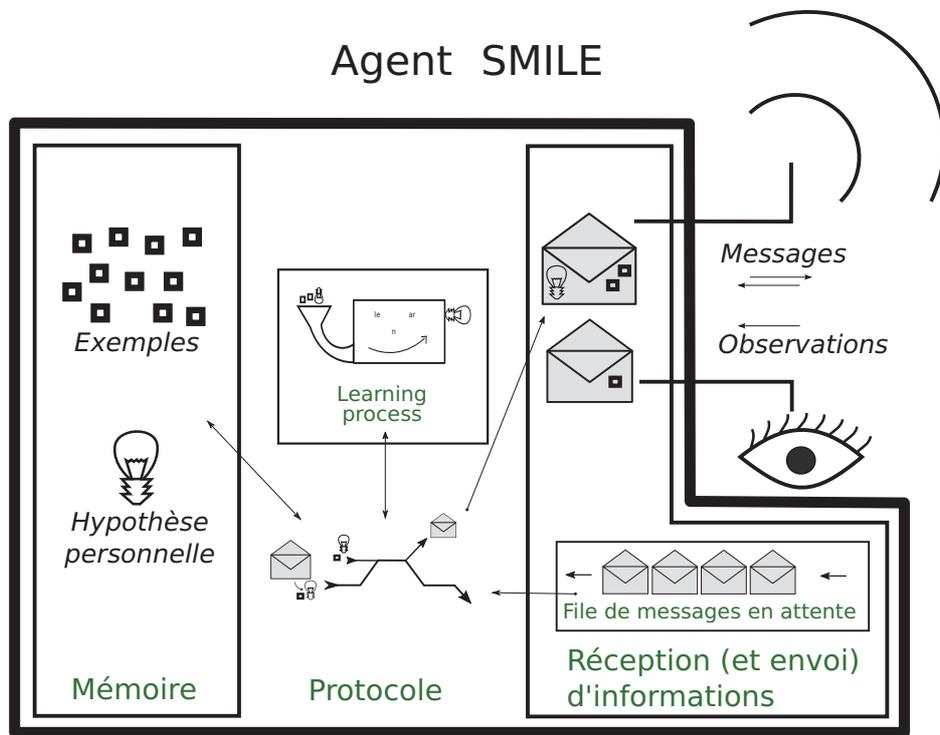


FIGURE 3.1 – Structure d’un agent incluant le protocole SMILE. Le protocole peut accéder à toutes les fonctionnalités de l’agent selon les besoins de la procédure spécifique au traitement de chaque message.

- SMA-cons (H_p) : les agents critiques remplacent leur hypothèse par H_p lors de la réception du message de SMA-consistance.

Un agent prend un rôle d’apprenti uniquement le temps d’une révision globale. Lorsque celle-ci est terminée, les agents ont tous la même probabilité de recevoir un nouvel exemple, si celui ci déclenche une révision ils peuvent alors devenir apprenti à leur tour (figure 3.5).

3.1.2 Évaluation d’un apprentissage

On définit *la justesse* de l’apprentissage d’un agent comme la probabilité que son hypothèse classe correctement un prochain exemple (en considérant le tirage d’un nouvel exemple comme une variable indépendante et identiquement distribuée). Cette justesse est évaluée sur un échantillon d’exemples de test dont on sait a priori lesquels sont positifs et lesquels sont négatifs.

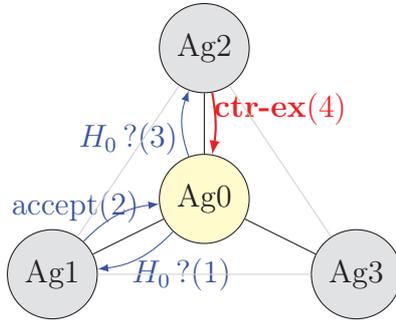


FIGURE 3.2 – Propositions successives d’une hypothèse H_0 à $Ag1$ qui accepte puis $Ag2$ qui possède et envoie un contre-exemple à H_0 , l’hypothèse doit être révisée avant de recommencer les propositions

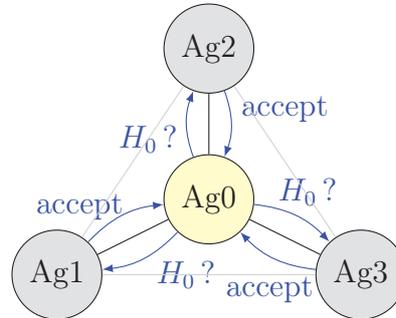


FIGURE 3.3 – Propositions toutes acceptées, l’hypothèse est SMA-consistante, la révision globale initiée par $Ag0$ se termine

La justesse mesurée correspond à la proportion de réponses correctes sur cet échantillon ; exemples positifs acceptés et exemples négatifs rejetés. Avec le protocole SMILE décrit précédemment les agents ont toujours tous la même hypothèse en fin de révision globale donc mesurer la justesse d’un agent revient à mesurer celle du SMA. On peut envisager des contraintes variées selon l’application de ce protocole. Le temps nécessaire à l’apprentissage peut être limité par les temps de communications ou encore par le rythme d’arrivée des exemples avant de considérer le temps d’exécution du protocole interne à chaque agent, lui même dépendant de la machine qui le supporte. On choisira donc de s’affranchir d’une mesure en fonction du temps et de mesurer l’évolution de la justesse du SMA en fonction du nombre total d’exemples reçus par le SMA. La notion de *rapidité d’apprentissage* ne correspond donc pas à avoir une hypothèse précise en peu de temps, mais en peu d’exemples. C’est par cette mesure de justesse que nous estimerons si un SMA tire parti des exemples reçus plus ou moins efficacement qu’un autre.

3.1.3 Résultats du protocole SMILE

Bien que chaque agent ne dispose pas de tous les exemples d’apprentissage fournis au SMA, le protocole SMILE permet à un SMA d’atteindre la même justesse d’apprentissage et la même rapidité d’apprentissage qu’un agent seul disposant de tous les exemples. Pour certains problèmes d’apprentissage comme le Multiplexer 11 (*M11*) ou encore le ou exclusif (*XOR*) (voir section 2.1.3) le protocole SMILE peut même être plus rapide qu’un agent

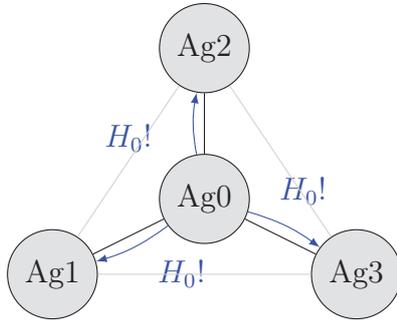


FIGURE 3.4 – Ag_0 ayant obtenu une nouvelle hypothèse SMA-consistante il en informe ses voisins. Aucune réponse n'est attendue et il n'est plus apprenti.

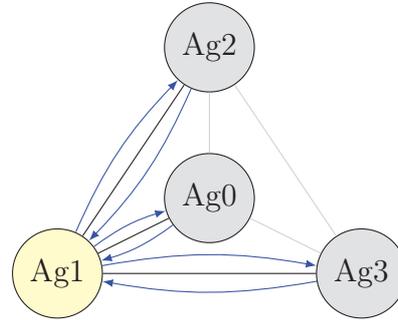


FIGURE 3.5 – Le rôle d'apprenti peut être pris par un autre agent (ici Ag_1) lors d'une autre révision globale.

seul qui recevrait tous les exemples. La figure 3.6 représentant le cas d'un ensemble de 50 agents reliés en Clique (le graphe des communications possibles entre agents contient une arête entre toute paire de sommet) apprenant la formule du M11 montre un tel résultat. Certaines explications complémentaires sont présentées dans l'article introduisant SMILE [BES07], ainsi que d'autres [BSE10] notamment concernant ce gain d'efficacité par rapport à un agent seul dans des cas d'apprentissage sans erreur de problèmes booléens difficiles.

Le protocole SMILE peut s'adapter à d'autres domaines que l'apprentissage de concept. Une de ses application concerne l'apprentissage actif de modèles d'actions présenté par Rodrigues et al à travers l'application IRALe [RSBR14]. Au sein de cette application, les agents mémorisent les exemples qui leur ont servi à réviser leur hypothèse. Ils peuvent ensuite s'en servir pour critiquer les hypothèses qui leur seront proposées. Chaque agent bénéficie ainsi de l'expérience des autres pour affiner son modèle d'action.

3.1.4 Variantes

Il est possible de modifier les conditions d'apprentissage par des variantes du protocole SMILE.

Individualisme : i-SMILE

On peut considérer que les agents doivent former leur hypothèse personnelle eux-même soit parce qu'ils ne font pas confiance aux autre soit parce que l'on considère qu'ils doivent faire le raisonnement complet conduisant à leur

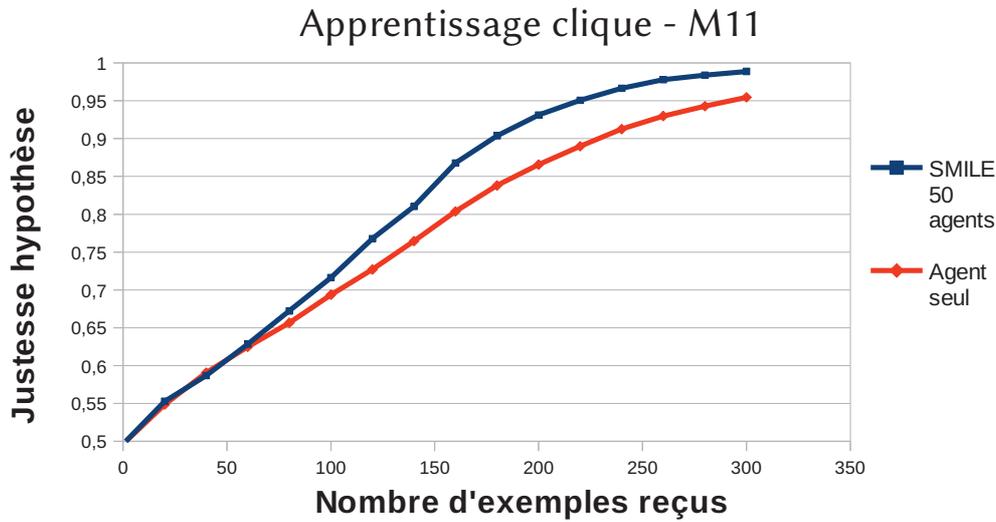


FIGURE 3.6 – Évaluation de l’apprentissage d’un SMA par la justesse de l’hypothèse du dernier appreni en fonction du nombre d’exemples reçus par le SMA. Cas concluant de 50 agents reliés par le protocole SMILE comparativement à un agent seul pour le problème du multiplexer11.

hypothèse. Il s’agit alors d’une version individualiste de SMILE, i-SMILE, qui n’utilise pas le processus d’adoption. Il est possible d’y adjoindre un moyen de resynchronisation afin de rétablir la SMA-consistance des hypothèses personnelles des agents ici potentiellement différentes les unes des autres. Les agents doivent alors re-proposer régulièrement leur hypothèse actuelle pour vérifier que les nouveaux exemples ne la contredisent pas ou la réviser si nécessaire.

Comme on peut s’y attendre, la version i-SMILE est moins performante en ce qui concerne la justesse. La resynchronisation permet de rétablir la justesse, mais au cout de nombreux échanges de messages supplémentaires ainsi qu’un plus grand nombre d’opérations élémentaires effectuées par agent, mais également une plus grande redondance des exemples en mémoire.

Broadcast

Principe du broadcast Un protocole avec *broadcast* a été proposé pour des agents tous connectés par Bourgne et al [BSE10] permettant à chaque agent de proposer son hypothèse à tous ses voisins en même temps. Les modifications par rapport au protocole de référence sont que l’appreni attend toutes les réponses avant d’en tenir compte. En cas de contre-exemple, il

doit faire une révision qui peut tenir compte de plusieurs exemples supplémentaires, au lieu d'un seul comme dans le cas séquentiel. Malgré une légère perte d'efficacité dans certains cas, le protocole assure la SMA-consistance des hypothèses des agents, au sein d'une Clique, à la fin de chaque révision globale, avec le mécanisme d'adoption.

Réponse maximale et oubli Comme un agent utilisant le broadcast peut tenir compte de plusieurs contre-exemples à la fois, il nécessite moins de révisions avant d'atteindre la SMA-consistance. On peut choisir d'augmenter ce phénomène avec l'option de *réponse maximale*. Tout agent critique renvoie alors l'ensemble de ses contre-exemples plutôt que de se limiter au premier qu'il trouve dans sa mémoire. Le broadcast, d'autant plus avec réponse maximale, augmente le nombre d'exemples transmis, ce que l'on peut vouloir éviter, soit pour des raisons d'intimité des agents soit pour l'impact sur la redondance des exemples en mémoire.

Il est toujours possible de limiter cette redondance par un oubli sélectif des exemples. Afin de conserver un exemplaire de chaque exemple en mémoire, ils sont tagués internes si l'agent les obtient de son environnement ou externes s'ils proviennent d'un autre agent. L'*oubli* des exemples externes permet de réduire la redondance des exemples en mémoire. Afin d'assurer la convergence du protocole il est nécessaire d'attendre la fin d'une révision globale pour effectuer un oubli.

D'après Bourgne et Al [BSE10] ce qui explique le gain de justesse de SMILE par rapport à un agent seul sur des problèmes difficiles est un équilibre entre profiter du travail réalisé et explorer plus l'espace d'hypothèses. On fera référence à cette différence de justesse, entre un agent seul et un apprentissage multi-agents, à nombre d'exemples d'apprentissage fixé, par l'appellation *gain exploratoire*. Certains critères peuvent faire varier l'apprentissage entre la stabilité et l'exploration. Peu d'agents, le broadcast, la réponse maximale et la conservation de tous les exemples en mémoire sont en faveur d'une exploration faible en partageant rapidement un maximum d'informations. Un nombre d'agents plus grand, l'oubli, la communication un à un et la réception d'un seul exemple à la fois favorisant des situations d'apprentissages diverses et donc une plus grande exploration et variabilité.

Dans ces conditions, le compromis entre exploration et exploitation bénéficie particulièrement à la version SMILE, broadcast, réponse maximale avec adoption et oubli. Il a une plus grande vitesse d'apprentissage et un

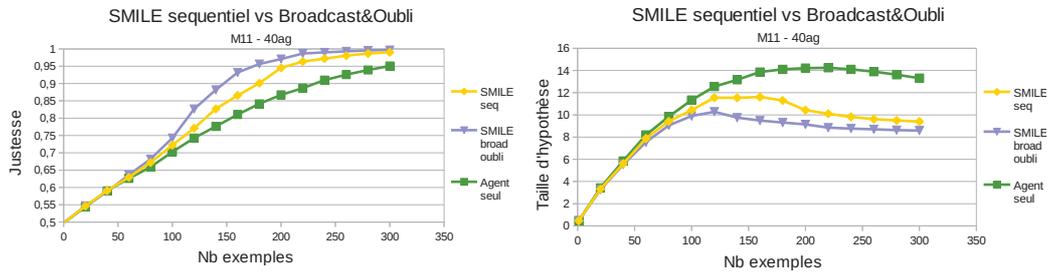


FIGURE 3.7 – La version broadcast et oubli de SMILE permet d’améliorer la qualité des hypothèses formées avec une meilleure justesse et taille plus courte.

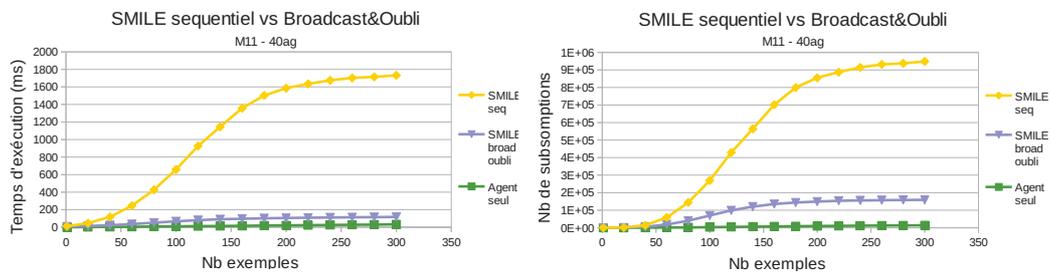


FIGURE 3.8 – La version broadcast et oubli de SMILE permet d’améliorer les coûts de l’apprentissage avec moins de calculs et de temps.

gain exploratoire de justesse conséquent avec des problèmes d’apprentissage difficiles.

L’oubli qui favorise l’exploration, peut donner lieu à la destruction d’une partie de l’hypothèse lors de la réception de contre-exemples négatifs. Ceci peut être néfaste lorsqu’un apprenti reçoit les contre-exemples un à un de ses nombreux voisins au risque de détruire totalement son hypothèse avant de commencer à la reconstruire. Le broadcast réponse maximale permet de la reconstruire immédiatement sans poursuivre une exploration trop lointaine.

De toute cette partie limitée à des agents en Clique, c’est-à-dire sans contraintes de communication, la version de SMILE avec broadcast et oubli est la meilleure actuelle. Elle améliore la qualité d’apprentissage tout en diminuant les coûts. La figure 3.7 montre le gain de vitesse d’apprentissage ainsi qu’une taille moyenne d’hypothèse plus petite. A justesse équivalente, une hypothèse plus courte est plus simple à interpréter. Cette version diminue également les coûts (figure 3.8) tant en ce qui concerne le temps d’exécution

que les calculs effectués et le nombre d’envois de messages.

3.2 Adaptation séquentielle au réseau

3.2.1 Transition vers le réseau

On peut imaginer toutes sortes d’ensembles d’agents et nombre de circonstances limitant les communications entre eux. Un réseau physique par exemple tel que des compteurs électriques pourra n’être câblé, en fonction de la distance, qu’entre agents les plus proches tandis que pour un réseau social, la proximité prendra la forme de relations personnelles ou professionnelles. L’analyse de phénomènes d’apprentissages collectifs dans un réseau n’est pas nouvelle. Zollman dans sa thèse *Network epistemology* [Zol07] montre des effets intéressants d’influence de la structure du réseau sur l’apprentissage individuel comme global.

Comment envisager l’apprentissage collaboratif composé d’agents autonomes, vu précédemment dans le cas d’une clique, lorsque l’on limite les liaisons entre agents par un réseau de communications ?

En faisant le choix de conserver la même stratégie, chaque agent A_a contacte alors seulement ses voisins A_c et cherche à s’assurer de la consistance de son hypothèse avec ses propres exemples et ceux de ses voisins directs. On appellera cette propriété *groupe-consistance*. Bien que cette approche ne permette plus de garantir l’obtention d’hypothèses SMA-consistantes, on peut se demander à quel point la qualité de l’apprentissage est pénalisée par le manque de liens de communications. Ceci sera développé dans le chapitre 4. On cherche également un moyen de rétablir la propriété de SMA-consistance au moyen de mécanismes de propagation. Les protocoles Relay et Variable-Root présentés ci-après sont tirés des articles de Bourgne et al [BES09] [BEMS08] qui proposent une adaptation de SMILE aux réseaux. Ces protocoles ont été conçus du point de vue de l’apprenti qui cherche à retrouver le contact avec tous les autres agents comme dans le cas bien étudié de la clique. L’apprenti prend dans ce cas là un rôle prépondérant tandis que les autres agents semblent devenir passifs le temps d’une révision globale. Cette vision des protocoles réseau ayant évolué, les protocoles réseau seront directement présentés comme le résultat de comportements individuels de chaque agent, ce qui respecte mieux la propriété d’autonomie recherchée.

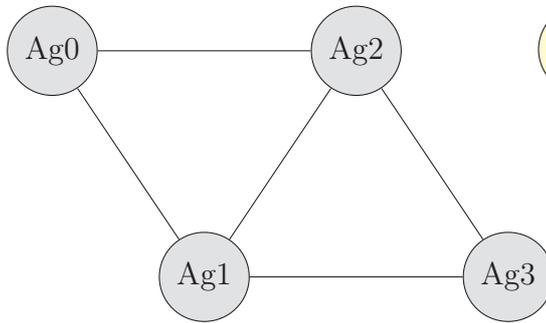


FIGURE 3.9 – Un graphe G de 4 agents

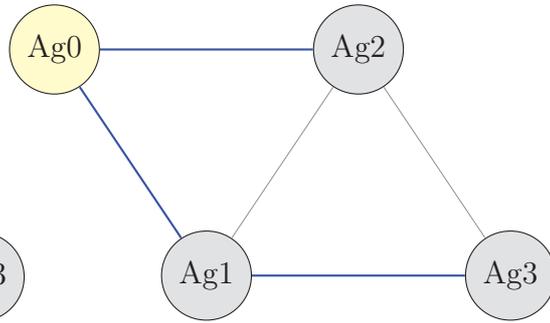


FIGURE 3.10 – Chemin d'une hypothèse proposée par Ag_0 dans G

3.2.2 Protocoles réseaux séquentiels

Déléguer pour simuler une clique : Relay

Une première idée est de se dire "Si je peux faire une révision de groupe avec mes voisins, je peux alors demander à mes voisins de poser la question à leurs voisins". Pour une hypothèse proposée il suffit alors que chaque agent réponde une et une seule fois à la question. Le chemin parcouru par une hypothèse entre sa proposition et son acceptation globale dessine alors un arbre couvrant du graphe initial.

On choisira arbitrairement un parcours du graphe correspondant à un arbre couvrant créé en largeur d'abord plutôt qu'en profondeur avec l'idée que l'on préfère que l'information fasse le plus court trajet possible entre celui qui l'envoie et celui qui la reçoit. On considère toujours un objectif d'apprentissage fixe dans le temps, dont les exemples sont tous sans erreurs, indépendants et identiquement distribués. Certains prolongements pourraient remettre en cause certaines de ces propriétés, rendant la structure en largeur d'autant plus importante. Il existe des protocoles similaires basés sur des arbres construits en profondeur d'abord.

Le protocole Relay peut également être défini par les différents types de messages échangés et leurs significations. Ils sont répertoriés ci-après. Les messages marqués (+) sont nouveaux par rapport à SMILE.

Protocole Relay :

- `propose(hypothèse)` : demande à l'agent s'il a des contre-exemples.
- `(+)weak-accept` : aucun contre-exemple de l'agent.

- (+)**strong-accept** : aucun contre-exemple de la branche.
- (+)**direct-accept** : réponse d'un agent qui dépend d'une autre branche (possède déjà l'hypothèse proposée).
- (+)**propagate** : ordre de demander dans toute la branche s'il y a des contre-exemples.
- **counter-example** (CE) : renvoie un contre-exemple.
- **SMA-cons**(h_p) : indique que l'hypothèse est SMA-consistante.

On considère toujours des agents avec une mémoire des exemples internes et externes reçus et d'une hypothèse. On propose de se ramener au problème déjà résolu de l'apprentissage en clique en permettant aux agents, dont le réseau de communication forme un graphe connexe G_c , de transmettre les propositions et les réponses à leurs voisins afin d'atteindre tous les agents du graphe. On souhaite que tous les agents répondent une et une seule fois à la question "l'hypothèse H est-elle consistante avec tes exemples" sous-entendue par un message **propose**(H). Pour chaque hypothèse H la question va parcourir le graphe en formant dynamiquement un arbre couvrant G_{ac} de G_c . Pour cela, le temps d'une révision, les agents vont faire une différence temporaire entre leurs voisins selon trois catégories : le père qui a posé la question, les enfants dont on attend la réponse et les étrangers qui dépendent d'une autre branche avec lesquels il n'y a plus d'arête dans G_{ac} . L'arbre couvrant formé va dépendre de l'apprenti qui initie la révision de manière à avoir des chemins courts entre l'apprenti et chaque agent. Afin de pouvoir proposer H à ses enfants et savoir s'il a déjà répondu à la question, **un agent autre que l'apprenti doit adopter l'hypothèse qu'il valide par un weak-accept la première fois qu'on lui propose et en mémoriser le père**. Deux autres formes de **accept** existent, **strong-accept** permet à un agent d'indiquer à son père l'absence de contre-exemple dans toute sa branche et pas uniquement dans sa propre mémoire, et **direct-accept** permet à un voisin étranger qui a déjà été consulté d'indiquer qu'il dépend d'une autre branche. Dans l'exemple 3 d'une révision sans contre-exemple (représenté figure 3.10), $Ag1$ a pour père $Ag0$, pour fils $Ag3$ et pour voisin étranger $Ag2$. Si un agent répond par un contre-exemple CE , celui ci est immédiatement remonté de fils à père dans la branche jusqu'à l'apprenti qui va réviser H en tenant compte de CE et recommencer ses propositions depuis le début. Le message **SMA-cons**(H) final est facultatif il permet d'indiquer à tous les agents que la révision globale est terminée, mais comme ceux-ci ont déjà tous adopté H , ils ne réagissent pas au message autrement qu'en le propageant.

Exemple 3 Révision globale sans contre-exemple.

Dans l'exemple représenté figure 3.10 un agent *Ag0* propose une hypothèse *H* à son voisin *Ag1* qui répond par un **weak-accept** puis son voisin *Ag2* qui répond également par un **weak-accept**. *Ag0* ayant consulté tous ses voisins sans obtenir de contradiction, il demande en premier à *Ag1* de propager *H*. *Ag1* mémorise que la demande vient de *Ag0* en le considérant comme son père. *Ag1* propose alors *H* à tous ses voisins qui ne soient pas son père successivement. On souhaite qu'un agent ne fasse pas la vérification plusieurs fois donc il adoptera l'hypothèse qui lui est proposée au moment où il renvoie un **weak-accept**. *Ag2* ayant déjà répondu à *Ag0*, il a adopté *H* et répond à *Ag1* par un **direct-accept**. *Ag1* sait alors que *Ag2* fait partie d'une autre branche et qu'il ne doit pas lui demander de propager. *Ag1* propose *H* à *Ag3* qui accepte (**weak-accept**). Ayant l'avis de tous ses voisins, *Ag1* veut propager, il ne contacte ni son père ni ses voisins qui dépendent d'une autre branche. *Ag1* demande de propager à *Ag3* qui a adopté *H*. *Ag3* contacte son seul voisin qui ne soit pas son père, *Ag2* qui répond une nouvelle fois par un **direct-accept**. *Ag3* peut répondre pour toute sa branche par un **strong-accept** à *Ag1*. *Ag1* synthétise également les réponses de sa branche (**direct-accept** et **strong-accept**) et fait remonter le **strong-accept** à *Ag0*. *Ag0* demande alors à *Ag2* de propager. *Ag2* n'obtient que des **direct-accept** de *Ag1* et *Ag3* et remonte un **strong-accept** à son père *Ag0*. *Ag0* sait alors que son hypothèse a parcouru le graphe entier sans rencontrer de contre-exemple et envoie un message de SMA-consistance à ses voisins qui le propagent à leurs enfants.

Transmettre le rôle d'apprenti : Variable-Root

En vertu de sa qualité d'autonomie, l'on souhaite d'un agent qu'il apprenne toujours au mieux avec les informations dont il dispose. Dans le cas précédent du protocole Relay, si un intermédiaire reçoit un contre-exemple, il se contente de transmettre l'information, mais ne s'en sert pas. Du point de vue de l'agent il serait plus intéressant d'apprendre de ce contre-exemple immédiatement plutôt que de s'en remettre à l'apprenti qui a initié la révision et qui se trouve potentiellement loin. D'un point de vue réseau il semble également plus intéressant de faire une révision le plus tôt possible pour s'épargner des messages intermédiaires.

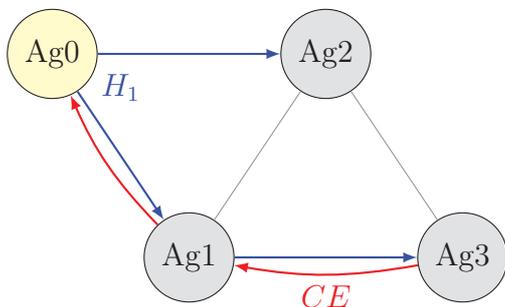


FIGURE 3.11 – Protocole Relay, retour d'un contre-exemple

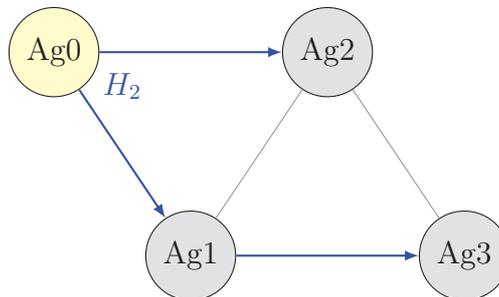


FIGURE 3.12 – Protocole Relay, révision de H_1 en H_2 puis proposition de H_2

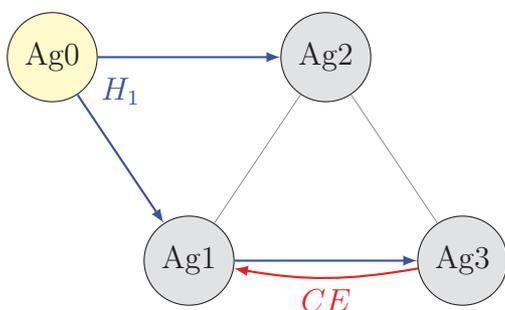


FIGURE 3.13 – Protocole Variable-Root, retour d'un contre-exemple

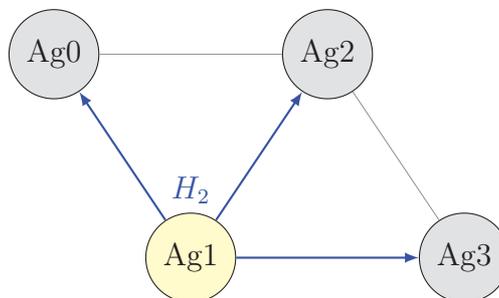


FIGURE 3.14 – Protocole Variable-Root, changement d'apprenti et révision de H_1 en H_2 puis proposition de H_2

Comparaison des protocoles Variable-Root et Relay lors de la réception d'un contre-exemple par un agent intermédiaire.

Le protocole Variable-Root propose donc une petite modification par rapport au protocole Relay, le changement de rôle pendant une révision globale. Lorsqu'un agent reçoit un contre-exemple il révisé H lui-même en H_r , devenant ainsi le nouvel apprenti (à la place de l'apprenti) et la racine du nouvel arbre couvrant qui va se construire sur le chemin de H_r à la place de l'arbre de H . En recevant un **propose** (d'une hypothèse qui n'est pas la sienne) un agent met à jour son père et s'il répond par un **accept**, il met également à jour son hypothèse. L'agent déterminera si les autres voisins sont ses fils ou des étrangers après la réception d'un **propagate**, lorsqu'il recevra la réponse au **propose** qu'il va envoyer à chacun d'eux. On reprend les conditions de l'exemple 3 où Ag_3 dispose d'un contre-exemple à l'hypothèse H_1 proposée par Ag_0 pour illustrer la différence d'exécution précédemment décrite des protocoles Relay et Variable-Root sur les figures 3.11 à 3.14.

Le protocole Relay a permis de rétablir la SMA-consistance des hypothèses de tous les agents après l'arrivée de tout exemple dans les réseaux de communication connexes. Le protocole Variable-Root permet, de plus, de rattraper les performances de la clique concernant la vitesse d'apprentissage. La lecture plus approfondie de l'article les présentant [BES09] donne plus de détails sur ces protocoles et leurs résultats.

Les chapitres suivants concernent mes apports personnels. On étudiera dans le chapitre 4 l'influence de la topographie du réseau sur la justesse de l'apprentissage limité au voisinage. Le chapitre 5 permet d'expliquer les limites en termes d'apprentissage en parallèle des protocoles réseaux précédemment présentés. Le chapitre 6 répond à ce problème avec un nouveau protocole : Waves.

Chapitre 4

Apprentissage d'un réseau dont les communications sont limitées au voisinage immédiat

Nous cherchons ici quelles caractéristiques des graphes ont une influence lors d'un apprentissage basé sur le protocole SMILE de référence qui limite les communications aux voisins immédiats [VBS17a]. On présente tout d'abord le fonctionnement de SMILE dans un contexte de réseau, mais sans propagation ainsi que la nature des informations qui sont échangées lors de l'apprentissage. Les résultats sont ensuite constitués d'une mise en lumière progressive des paramètres importants de la structure d'un réseau. L'étude de l'influence relative de la propagation des exemples et des hypothèses vient compléter celle de la forme des réseaux.

4.1 Variables de l'étude

Dans ce chapitre nous considérons diverses formes de graphes afin d'identifier quelles caractéristiques ont une influence sur la vitesse d'apprentissage. Par ailleurs deux formes d'informations sont échangées lors des communications, les exemples qui sont des faits sans erreur, mais dont il faut un grand nombre pour former une hypothèse de qualité et des hypothèses qui sont le résultat d'un travail préalable de résumé et de généralisation. L'information contenue dans une hypothèse est l'ensemble des exemples qu'elle classe correctement.

4.1.1 Exemple détaillé de révision de groupe

Une grande différence entre le fonctionnement en clique et celui en réseau avec le même protocole SMILE de référence est que malgré le mécanisme d'adoption, il n'est plus possible d'assurer des hypothèses SMA-consistante à la fin de chaque révision de groupe. Un agent apprenti ne propose son hypothèse qu'à son voisin immédiat et tout autre agent à distance deux ou plus n'est ni consulté ni informé du résultat de la révision de groupe. En clique, une révision de groupe inclut nécessairement l'ensemble des agents et est donc systématiquement une révision globale. L'exemple 4 repris par la figure 4.1 illustre une de ces révision de groupe dans le cas d'une disposition en ligne de 4 agents.

Exemple 4 Révision de groupe.

On se place dans un cas de quatre agents alignés dont l'objectif est d'apprendre le concept "A ou B" à partir d'exemples positifs (appartenant au concept) et négatifs (n'appartenant pas au concept). L'hypothèse courante d'un agent doit accepter les exemples positifs (e^+) et rejeter les exemples négatifs (e^-) connus par l'agent pour être consistante avec ces exemples. Après que *Agent*₁ et *Agent*₃ aient reçu respectivement les exemples positifs $e_1^+(A \wedge \neg B \wedge C)$ et $e_2^+(A \wedge B \wedge \neg C)$ puis révisé leurs hypothèses en conséquence (respectivement H_1 puis H_2), *Agent*₂ reçoit un exemple positif $e_3^+(\neg A \wedge B \wedge C)$ (figure 4.1). Il ajoute e_3^+ à sa mémoire d'exemple et constate une contradiction avec son hypothèse courante. A la suite de sa révision il forme l'hypothèse H_3 qu'il soumet à la critique de son voisin *Agent*₁. Celui ci dispose d'un contre-exemple e_1^+ non consistant avec H_3 et l'envoie comme réponse à *Agent*₂. A l'issu d'une deuxième révision, qui tient compte de e_1^+ ajouté en mémoire, *Agent*₂ propose sa nouvelle hypothèse H_4 successivement à *Agent*₁ et *Agent*₃, ses voisins, qui l'acceptent faute de contre-exemple à renvoyer. L'hypothèse H_4 étant cohérente avec la mémoire de *Agent*₂ et ses voisins, il a terminé sa révision de groupe. *Agent*₁ et *Agent*₃ étant voisins de *Agent*₂, ils adoptent sa nouvelle hypothèse H_4 . Par le mécanisme de mémorisation, *Agent*₂ conserve en mémoire son exemple externe e_1^+ en plus de ses exemples internes.

4.1.2 Variables du protocole SMILE

Dans le cas de communications limitées, une hypothèse peut perdre, lors d'une révision de groupe, une partie de l'information qu'elle contenait comme

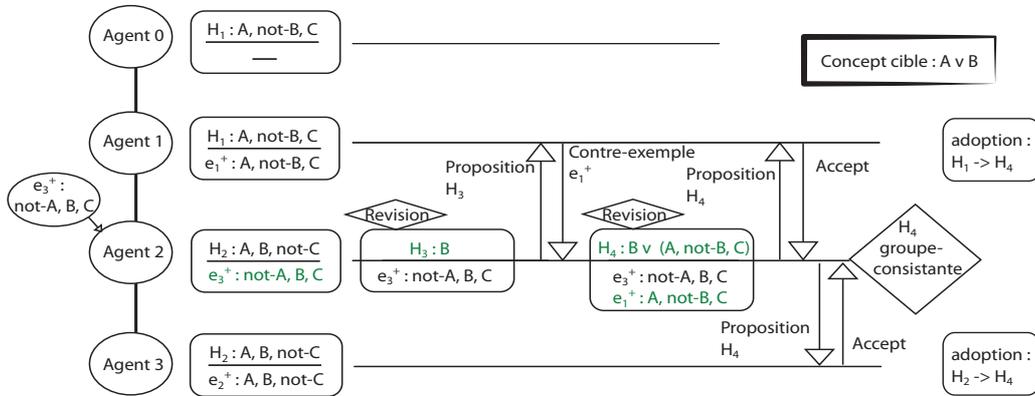


FIGURE 4.1 – Exemple d’apprentissage collectif local dans un réseau de 4 agents alignés avec le protocole SMILE de référence (adoption et mémorisation).

dans l’exemple 5. La formation d’une hypothèse par plus petite généralisation fait d’elle un condensé d’information par rapport à la transmission de tous les exemples, mais il y a un risque qu’une partie de cette information se perde par la suite. On notera par ailleurs qu’il existe des phénomènes de propagation tant pour les exemples (exemple 6) que pour les hypothèses (exemple 7). Les exemples se propagent grâce à l’envoi de contre-exemples par un agent critique à un agent apprenti qui les mémorise en tant qu’exemples externes. Dans le cas des hypothèses, elles sont propagées d’un apprenti vers les critiques qui l’entourent au moyen de l’adoption. À plus long terme, une partie de l’information d’une hypothèse est conservée lors de sa prochaine révision et peut être transmise plus loin, mais ce n’est pas garanti dans une structure de réseau.

Définition 4. On définit une **hypothèse fille** H_f comme le résultat de la révision d’une hypothèse H_m . H_m est réciproquement appelée **hypothèse mère** dès qu’il existe une hypothèse fille formée à partir de la révision de H_m .

H_f hérite d’une partie de l’information de H_m dans le sens où elle sépare un certain nombre d’exemples de la même façon.

Exemple 5 Perte d’information lors d’une révision : dans la disposition de l’exemple 4 de quatre agents 0 à 3 en ligne, si l’*Agent*₂ fait une révision de groupe, il tient compte des exemples des agents 1, 2 et 3. A la fin de la révision de groupe les trois agents

possèdent l'hypothèse H_d révisée grâce au mécanisme d'adoption. Si ensuite $Agent_1$ fait la révision de groupe suivante alors qu'il n'a pas en mémoire l'exemple e_1 détenu uniquement par l'agent 3. $Agent_1$ peut transformer H_d qui était consistante avec e_1 en H_n qui ne l'est plus. Lorsqu' $Agent_1$ propose H_n ni $Agent_0$ ni $Agent_2$ ne possèdent le contre-exemple e_1 et H_n ne sera pas rectifiée à l'issue de la révision de groupe.

Exemple 6 Propagation d'exemples : en utilisant le même graphe de quatre agents en ligne, alors que $Agent_3$ a reçu un exemple ex_1 , il se peut que $Agent_2$ fasse une révision de groupe au cours de laquelle $Agent_3$ lui envoie e_1 comme contre-exemple. Plus tard, lors d'une autre révision de groupe initié par $Agent_1$, $Agent_2$ envoie e_1 , qu'il a maintenant en mémoire, comme contre-exemple. Ainsi l'exemple se sera propagé d'abord à $Agent_2$ puis à $Agent_1$.

Exemple 7 Propagation d'hypothèses : en utilisant le même graphe de quatre agents en ligne, $Agent_1$ initie une révision de groupe à l'issue de laquelle les agents 0, 1 et 2 possèdent tous la même hypothèse H_1 . $Agent_2$ réalise la révision de groupe suivante à partir de H_1 pour former H_2 . $Agent_3$ en adoptant H_2 peut recevoir une partie de l'information de H_1 qui s'est propagée par hérédité et adoption.

En ayant ainsi conscience que l'information se trouve sous deux formes différentes, exemples et hypothèses, on peut se demander l'influence relative de la propagation de l'une ou l'autre. Il est possible de bloquer cette propagation pour en observer l'influence.

Concernant les exemples, ils sont transmis en réponse à un **propose** lorsqu'ils ont une valeur de contre-exemple. On peut faire en sorte que l'agent qui reçoit un contre-exemple ne le garde en mémoire que le temps de la révision de groupe. Alors que les exemples internes, reçus par un agent A directement de l'environnement, sont toujours gardés en mémoire, les exemples externes, provenant des agents voisins, dépendent du mécanisme de mémorisation. Il s'agira de **mémorisation** ou d'**oubli** des **exemples externes** selon que A les garde en mémoire ou non en fin de révision de groupe. L'oubli bloque la propagation des exemples.

Du côté des hypothèses, la propagation se fait par le mécanisme d'adoption. En bloquant le mécanisme d'adoption les agents critiques sont simplement consultés lors d'un processus de révision de groupe, mais ne tirent pas

parti du message final de groupe-consistance.

Le protocole SMILE de référence comprend la mémorisation des exemples externes et l'adoption des hypothèses groupe-consistantes, permettant ainsi la propagation de l'information sous forme d'exemples et d'hypothèses.

4.2 Résultats d'apprentissage limité au voisinage immédiat

Les agents n'ont plus tous la même hypothèse en fin de révision de groupe donc on ne peut plus évaluer la justesse du SMA avec la justesse d'un seul agent. Il existe plusieurs méthodes pour évaluer la justesse d'un SMA selon que l'on envisage un processus de décision de groupe tel que le vote ou uniquement des décisions individuelles. On choisit de mettre l'accent sur l'autonomie des agents en dissociant les phases d'apprentissage et de restitution. Un agent tient compte des connaissances des autres pour apprendre, mais il ne consulte personne d'autre lors de l'évaluation de son apprentissage. La justesse du SMA est dans ce cas la probabilité qu'un agent quelconque classe correctement le prochain exemple (par tirages indépendants et identiquement distribués). L'évaluation de la justesse du SMA sera alors la **justesse moyenne**, correspondant à la moyenne des justesses des agents.

Les SMA étudiés sont représentés par leur graphe de communication. On garde un nombre d'agents fixe en se plaçant à une échelle de 50 agents par graphe et on recherche alors l'influence de la topographie, présence et organisation des arêtes, sur l'apprentissage du SMA. Les résultats sont tous calculés sur une moyenne de 100 ou 200 séquences différentes d'ordre d'arrivée d'exemples d'un même problème d'apprentissage. On utilisera comme concept cible, une formule booléenne pour laquelle SMILE est efficace, le multiplexer11.

4.2.1 Des graphes similaires : Small-World

On s'intéresse en premier à une structure classique, les réseaux petits monde ou Small-World. Ils sont appelés ainsi pour signaler que tout sommet peut atteindre tout autre sommet en parcourant peu d'arêtes, ce qui correspond à une distance moyenne faible. On génère quelques Small-World par l'algorithme de Watts-Strogatz [Wat98]. Avec un nombre d'agents fixé à 50, il reste deux paramètres pouvant varier pour cet ensemble de graphe, le degré moyen k et la probabilité de modifier chaque arête p . Par exemple le

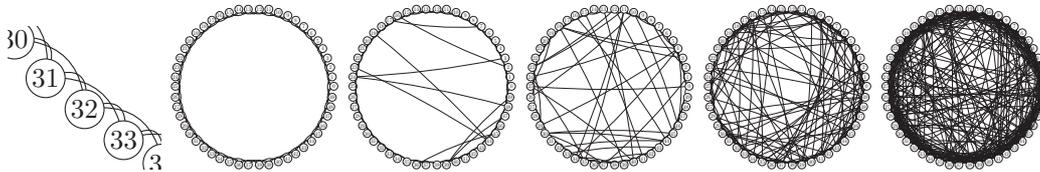


FIGURE 4.2 – Reg4(zoom-entier)-SmW04p01-SmW04p05-SmW08p05-SmW16p05

graphe SmW04p05 correspond à un graphe de type Small-World de degré moyen 4 et de probabilité 0,5. On rappelle également que le degrés moyen est proportionnel à la densité pour un nombre de sommets fixé, comme vu en section 2.3.1, donc on se référera indifféremment à ces deux notions dans ce chapitre.

En faisant varier le degré moyen, on fait varier la densité du graphe. Plus la densité est élevée plus le graphe considéré est proche d'une clique, le cas de référence. Les courbes d'apprentissages (figure 4.3) des graphes dont les moins denses sont représentés figure 4.2, confirment que plus la densité est élevée plus l'apprentissage du SMA est rapide au sens défini en 3.1.2. Cependant la densité n'est pas le seul facteur modifiant la justesse de l'apprentissage comme le montre la différence entre SmW04p01 et SmW04p05.

La variation de p (0 pour un graphe régulier et 1 pour un graphe aléatoire) en passant de 0,1 à 0,5 augmente le nombre d'arêtes coupant l'intérieur du cercle diminuant ainsi le diamètre et la distance moyenne. Nous allons voir l'influence de ces deux paramètres dans la partie suivante.

4.2.2 Des graphes de même densité

En ayant confirmé la forte influence de la densité, on cherche maintenant si la distance moyenne ou le diamètre d'un graphe expliquent entièrement les différences de rapidité d'apprentissage entre deux structures de graphe distinctes ayant même densité. On fixe la densité en prenant un lot de graphes de degré moyen 4 qui sont tous connexes, mais de diamètre et de distance moyenne variés. On choisit des graphes réguliers, d'autres de type pôles principaux-nœuds secondaires, des Small-Worlds en partie aléatoires et des graphes de tribus localement très denses. Les graphes Reg4 et le Donut (tore plat de longueur 10 et de largeur 5) sont des graphes réguliers dont tous les sommets sont placés équitablement. Les graphes de tribus ici sont formés de 10 cliques de 5 agents dont on retire une arête pour la placer entre deux cliques. Dans un cas on choisit une disposition des cliques en cercle pour MCluster5Circ et dans l'autre en étoile pour MCluster5Star.

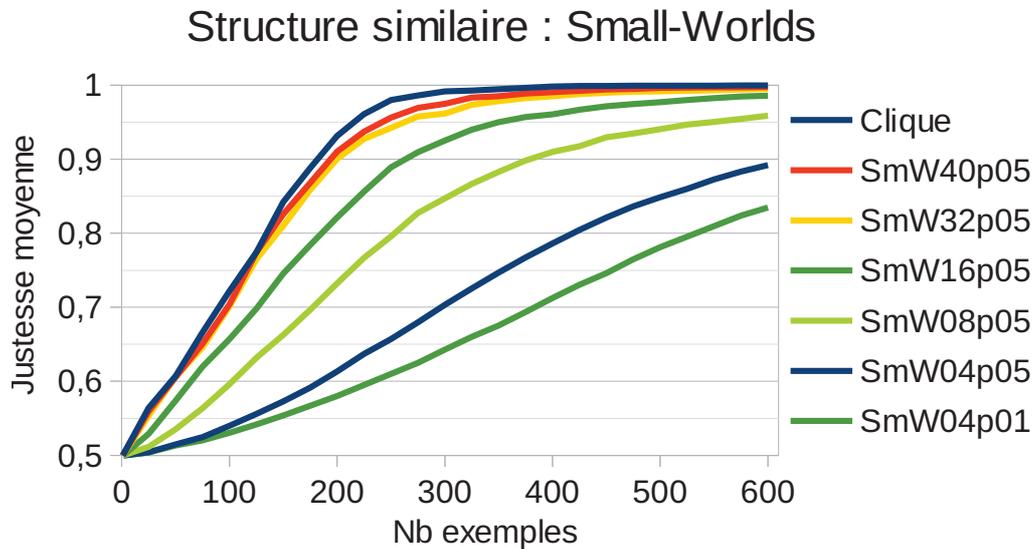


FIGURE 4.3 – Évolution de la justesse moyenne des hypothèses d’un SMA apprenant par le protocole SMILE selon diverses dispositions réseau de type Small-World.

On constate effectivement une certaine tendance telle que plus la distance moyenne est faible plus l’apprentissage est efficace et rapide. Au cours d’une révision de groupe l’information se déplace au plus d’une distance de une arête, d’un critique vers l’apprenti pour des exemple et de l’apprenti vers les critique pour l’hypothèse. Une distance plus grande à parcourir signifie plus de révisions nécessaires à la transmission d’une information. Cette tendance n’explique cependant pas totalement l’ordre des graphes ni la différence entre graphe de même distance moyenne (Bipôle, Tripôle, Roue). Pour cet ensemble de graphes la corrélation entre la justesse moyenne et la distance moyenne n’est que de $-0,77$ à 200 exemples. La tendance est la même pour le diamètre avec un corrélation un peu plus faible de $-0,74$. Ces valeurs de corrélation diminuent en valeur absolue dès que l’on prend un ensemble de graphe, faisant varier également la densité. Ces deux paramètres restent beaucoup plus significatifs que d’autres paramètres envisagés tels que les coefficients de clustering moyen ou globaux.

Nous souhaitons approfondir le cas du trio de graphes Roue-Bipôle-Tripôle qui bien qu’ayant même densité, diamètre et distance moyenne ne présentent pas les mêmes résultats en termes d’efficacité.

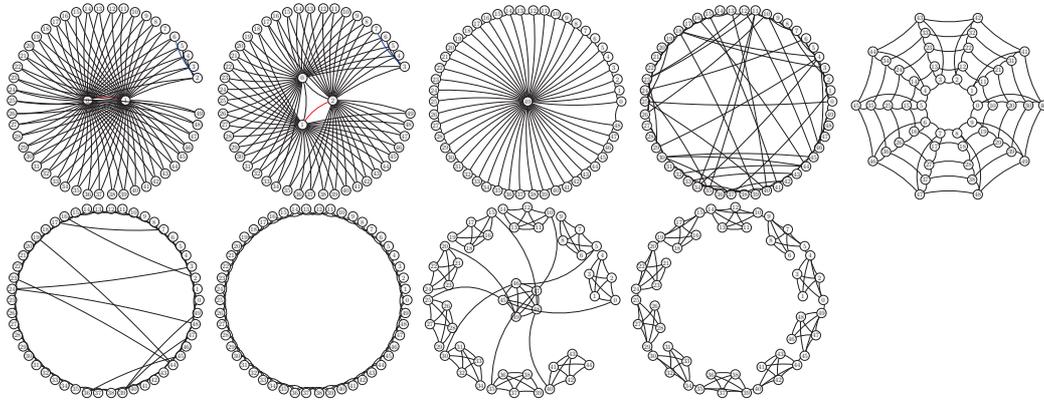


FIGURE 4.4 – Graphes de degré moyen 4 (distance moyenne) dans l'ordre de l'apprentissage le plus rapide au plus lent :
 Bipôle-Tripôle-Roue(1,92), SmW04p05 (2,95), Donut(3,78),
 SmW04p01(4,22), reg4(6,63), MCluster5Star(4,20), Mcluster5Circ(5,89)

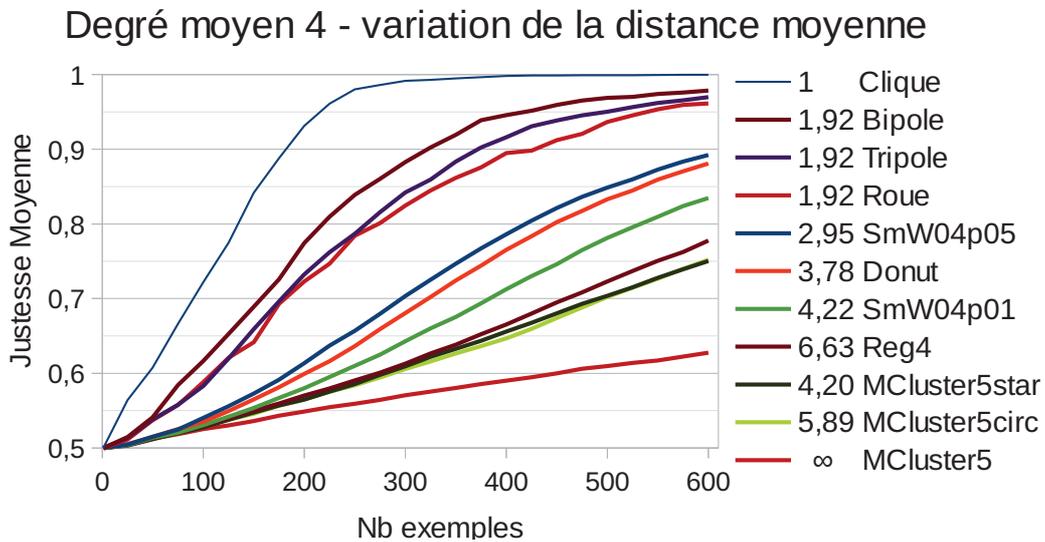


FIGURE 4.5 – Évolution de la justesse moyenne des hypothèses d'un SMA apprenant par le protocole SMILE selon diverses structures de réseau connexes de degré moyen 4. Le graphe non connexe MCluster5 et la clique sont présents en tant que références. La légende indique la distance moyenne et le nom des graphes.

4.2.3 Des graphes de même densité et distance moyenne, les multipôles

La distance moyenne est importante car toute distance représente le nombre minimum de révisions de groupe successives pour qu'une information aille d'un agent à un autre dans le SMA. On peut supposer que s'il y a plusieurs successions de révisions (chemins) possibles qui ont toutes cette longueur minimale (distance entre deux agents), l'information a d'autant plus de chances d'être transmise. L'information, sous la forme d'exemples et d'hypothèses, suit un chemin dans le graphe qui correspond à ces révisions successives. Le fait d'avoir plusieurs chemins alternatifs plutôt qu'un seul peut alors se mesurer en nombre de chemins minimaux de longueur n . Les chemins minimaux de longueur 1 correspondent aux arêtes qui sont déjà comptabilisées dans le paramètre de densité. La valeur suivante correspond donc aux chemins minimaux de longueur 2.

En reprenant le trio de Multipôles (Roue-Bipôle-Tripôle) on peut montrer qu'ils ont effectivement des différences au regard du nombre de chemins minimaux de longueur 2. Ces graphes ont en effet la même densité, ce qui avec un nombre d'agents fixe signifie même nombre d'arêtes et donc même nombre de chemins minimaux de longueur 1. Ils ont, de plus, le même diamètre (2) limitant les chemins minimaux à une longueur maximale de 2, mais ceux-ci sont plus nombreux dans le Bipôle (4520) que dans le Tripôle (3000) ou encore la Roue (2352). On peut voir dans le tableau 4.1 une corrélation entre la variation du nombre de chemins minimaux de longueur 2 et celle de la justesse.

On remarque par ailleurs que les liens entre les pôles et les liens entre nœuds secondaires apportent moins de chemins minimaux de longueur 2 que les liens entre les pôles et les nœuds secondaires.

nbr pôles	densité	diamètre	dist. moy.	nbr chem 2	justesse
1	0,08	2	1,92	2352	0,82
2	0,08	2	1,92	4520	0,88
3	0,08	2	1,92	3000	0,85

TABLE 4.1 – Le nombre de chemins minimaux de longueur 2 est corrélé à la justesse moyenne des hypothèses à 300 exemples lorsque les autres paramètres de densité diamètre et distance moyenne sont fixés.

On cherche à confirmer cette tendance sur un ensemble de graphes un

graphe (d=0,08)	diam.	dist. moy.	nbr chem 2	justesse
Reg4	13	6,63	300	0,77
MCluster5Star	7	4,20	188	0,75

TABLE 4.2 – Comparaison de deux graphes de degré moyen 4. Exemple de cas où le nombre de chemins minimaux de longueur 2 a plus d'importance que le diamètre ou la distance moyenne.

peu plus grand. On choisit les Multipôles de densité 0,2 pour lesquels on peut maintenir un diamètre de 2 de 1 à 9 pôles. De 1 à 5 pôles les nœuds principaux sont reliés à tous les nœuds secondaires donc ajouter un pôle augmente le nombre de chemins minimaux de longueur 2. Entre 5 et 9 pôles par contre, ajouter un pôle diminue le nombre de nœuds secondaires auquel chacun d'eux est relié ainsi que le nombre total de chemin minimaux de longueur 2. On retrouve exactement les mêmes variations au niveau de l'évolution de la justesse moyenne des SMA représentés par ces graphes. La corrélation entre nombre de chemins minimaux de longueur 2 et justesse moyenne pour ces graphes de 1 à 9 pôles atteint entre 0,945 et 0,97 selon le nombre d'exemples reçus auquel on effectue les mesures.

Évidemment cette corrélation diminue dès que l'on fait varier diamètre, distance moyenne ou densité et l'importance relative du nombre de chemins minimaux de longueur 2 par rapport à la distance moyenne reste indéterminée. A l'inverse on peut penser qu'à diamètre distance moyenne et nombre de chemins minimaux de longueur 2 équivalents, le nombre de chemins minimaux de longueur 3 puis 4 ... auront leur importance.

En général la distance moyenne a plus d'influence que le nombre de chemins minimaux de longueur 2, mais l'on trouve des exemples contraires comme le montrent le tableau 4.2 pour deux des graphes de degrés moyen 4 vus en section 4.2.2 et la figure 4.6 qui introduit deux nouveaux multipôles Multip21d02 et MultiP21d02sep.

4.2.4 Évolution de l'influence des paramètres au cours de l'apprentissage

Le décompte du nombre de chemins minimaux de longueur 2 explique la très grande majorité des variations de justesse de l'apprentissage, d'un SMA à un autre présentant un même nombre d'agents, une même densité et une même distance moyenne. La densité est prédominante sur le nombre de chemins minimaux de longueur 2 puisqu'un chemin de longueur 1 est plus intéressant que plusieurs chemins de longueur 2. Il reste difficile de quantifier

graphe (d=0,2)	diam.	dist. moy.	nbr chem 2	justesse
MultiP21	3	2,12	1416	0,77
MultiP21(sep)	4	2,34	4576	0,87

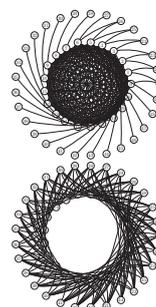


FIGURE 4.6 – Comparaison de deux multipôles de même densité et même nombre de pôles selon qu'ils sont reliés ou séparés. Exemple de cas où le nombre de chemins minimaux de longueur 2 est nettement plus important que le diamètre ou la distance moyenne.

l'influence de la distance moyenne par rapport à la densité et au nombre de chemins minimaux. On peut également se demander si cette influence varie en fonction de l'avancement de l'apprentissage.

On forme un ensemble de graphes constitué de plusieurs sous-ensembles et on calcule pour chaque paramètre sa corrélation avec la justesse à différents pas successifs de nombre d'exemples dans le SMA. Les sous-ensembles de graphes utilisés sont les Small-Worlds et les graphes de degrés moyen 4 présentés ci-avant, mais également le circuit(Reg2), deux Multipôles supplémentaires MultiP21d02 et MultiP21d02sep et un ensemble d'arbres dont l'étoile (Arb49). On remarque dans le tableau 4.3 que l'influence de la densité, bien qu'initialement beaucoup plus forte, diminue au fur et à mesure que l'apprentissage progresse, tandis que celle de la distance moyenne augmente. L'influence du diamètre est similaire à celle de la distance moyenne bien que toujours plus faible.

4.3 Propagation d'exemples et d'hypothèses

On mesure l'influence de la propagation des exemples et des hypothèses en comparant quatre versions du protocole SMILE selon qu'elles permettent ou bloquent les mécanismes de mémorisation et d'adoption. On Remarque tout d'abord que les effets de ces mécanismes diffèrent pour quelques graphes distincts puis on observe en réunissant les graphes ensemble que selon le protocole certaines caractéristiques de graphe sont plus influentes.

Nb Ex	degré moyen	diamètre	distance moyenne
100	0,848	-0,453	-0,557
200	0,787	-0,501	-0,606
300	0,693	-0,560	-0,668
400	0,613	-0,609	-0,716
500	0,559	-0,657	-0,761
600	0,517	-0,696	-0,797

TABLE 4.3 – Corrélacion entre différents paramètres de graphe et la justesse obtenue lors de plusieurs étapes successives de l'apprentissage. Le degré moyen, initialement plus influent, s'efface peu à peu devant la distance moyenne.

4.3.1 Effet théorique

Dès lors que les agents sont reliés en réseau connexe, mais pas en clique, ils bénéficient de la mémorisation des exemples externes, en particulier parce qu'elle rend des exemples accessibles pour un plus grand nombre d'agents, mais aussi comme économie de communication. Les agents bénéficient toujours de l'adoption d'hypothèse bien que celle-ci prenne ici, un sens plus limité. Dans une clique les décisions d'adoption ne sont prises qu'après consultation de l'ensemble des agents. Adopter une hypothèse dans une Clique permet de remplacer une hypothèse, qui n'est plus garantie SMA-consistante depuis l'arrivée de nouveaux exemples, par une autre qui est garantie. Dans un réseau avec le protocole de référence, la nouvelle hypothèse adoptée n'est pas toujours meilleure concernant le nombres d'exemples du système dont on soit certain qu'ils soient couverts. Par exemple au sein d'une étoile, une révision faite par l'agent central couvre les exemples de tous les agents, mais il adoptera néanmoins une hypothèse formée par un nœud secondaire qui ne couvre la mémoire que de deux agents.

4.3.2 Différents profils selon le diamètre

Lorsque l'on veut déterminer l'influence de la mémorisation et celle de l'adoption d'hypothèse on distingue deux cas qui correspondent à un diamètre égal à 2 d'une part et un plus grand diamètre d'autre part. On peut observer cette différence notable sur la figure 4.7 qui comprend le Tripôle de diamètre 2 et le Donut de diamètre 7. La Clique en tant que cas limite et le MultiP21d02 en tant que cas intermédiaire sont également fournis, figure 4.8.

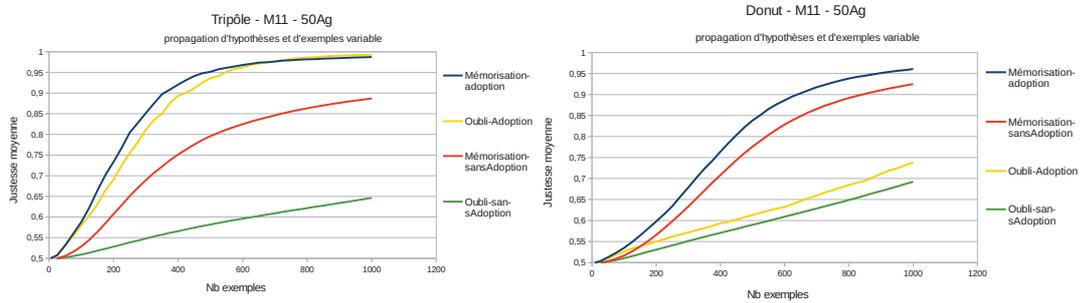


FIGURE 4.7 – Évolution de la justesse de l’apprentissage d’un même SMA en fonction de la propagation ou non des exemples (mémorisation-oubli) et des hypothèses (adoption-sansAdoption). Profils du Tripôle de diamètre 2 (gauche) et du Donut de diamètre 7 (droite).

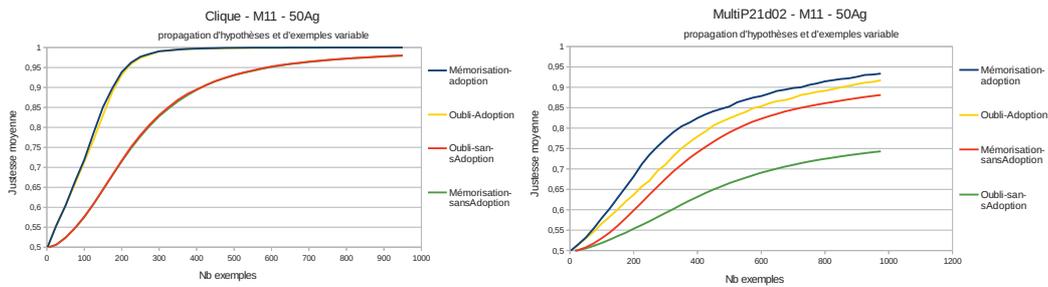


FIGURE 4.8 – Évolution de la justesse de l’apprentissage d’un même SMA en fonction de la propagation ou non des exemples (mémorisation-oubli) et des hypothèses (adoption-sansAdoption). Profils de cas limites, la Clique de diamètre 1 (gauche) et MultiP21d02 de diamètre 3 (droite).

Un diamètre très court tel que 2 implique l’existence de nœuds à degrés très élevés. Ces nœuds, appelés pôles, font des révisions de très bonne qualité, proches de révisions globales. Ils ont accès, grâce à leurs nombreux voisins, à la majorité des exemples du système. L’adoption permet de faire profiter régulièrement tous les agents d’une hypothèse proche de la SMA-consistance, la disponibilité des exemples pour les autres nœuds n’a alors pas beaucoup d’importance. Pour la Roue ou le Bipôle, qui présentent des pôles connectés à tous les autres agents sans exception, la mémorisation n’a plus aucune importance tant qu’il y a de l’adoption d’hypothèse. Pour le Tripôle par contre, les pôles ne sont reliés que au $2/3$ des nœuds secondaires et l’oubli pénalise alors légèrement l’apprentissage bien que le phénomène d’adoption soit prédominant. Si l’adoption n’est pas possible alors les pôles auront encore un grand

rôle à jouer en faisant leur révision puisque c'est l'occasion d'apprendre des exemples parmi les plus pertinents : ceux qui lui ont servi de contre-exemple. En l'absence de toute propagation les nœuds de faible degré n'ont accès qu'à très peu d'information et l'apprentissage moyen est lent.

Dans le cas de diamètre plus grand, il n'y a jamais de révision presque globale ni de moyen sûr d'obtenir d'hypothèse presque SMA-consistante. L'adoption a toujours en moyenne une influence bénéfique : les révisions d'hypothèses bien que pouvant dans de rares cas avoir une influence négative sur la qualité d'une hypothèse, sont plus souvent enrichissantes. La nouvelle hypothèse a une portée faible et peut parfois être oubliée sans avoir été réutilisée (exemple 8), mais elle tient toujours compte d'au moins un nouvel exemple. La mémorisation des exemples, est une progression certaine. Tout exemple appris par un nouvel agent (protocole avec mémorisation) ne peut être oublié et devient accessible à plus d'agents. Les exemples se propagent lentement mais sûrement.

Exemple 8 En considérant une ligne de quatre agents A_0 à A_3 et une révision de groupe de A_1 permettant à A_0 , A_1 et A_2 d'avoir comme nouvelle hypothèse H_1 . Si la prochaine révision est celle de A_2 il part de H_1 et bénéficie de l'adoption qu'il vient de faire, mais s'il effectue une révision plus tard, après A_3 , alors il aura une toute autre hypothèse de départ et n'aura eu aucun bénéfice de la révision de A_1 .

4.3.3 Adoption et mémorisation dépendent de différents paramètres

On réalise pour un ensemble de graphes divers une étude de leurs performances selon que l'on utilise ou bloque les mécanismes d'adoption et mémorisation.

Cas sans aucune mémorisation ni adoption

Dans ce premier cas, figure 4.9, on observe que la justesse obtenue selon le nombre d'exemples reçus ne dépend initialement que du degré moyen du graphe. L'approche apprenant-critique permet aux agents de tenir compte de leur exemples et ceux de leur voisins, ce qui justifie que, plus ils ont de voisins, plus ils ont d'exemples disponibles et, apprennent donc plus vite avec un degré moyen élevé. On remarque ensuite que l'apprentissage ralenti plus tôt pour une répartition inégale du degré. Par exemple, MultiP21d02 ne présente

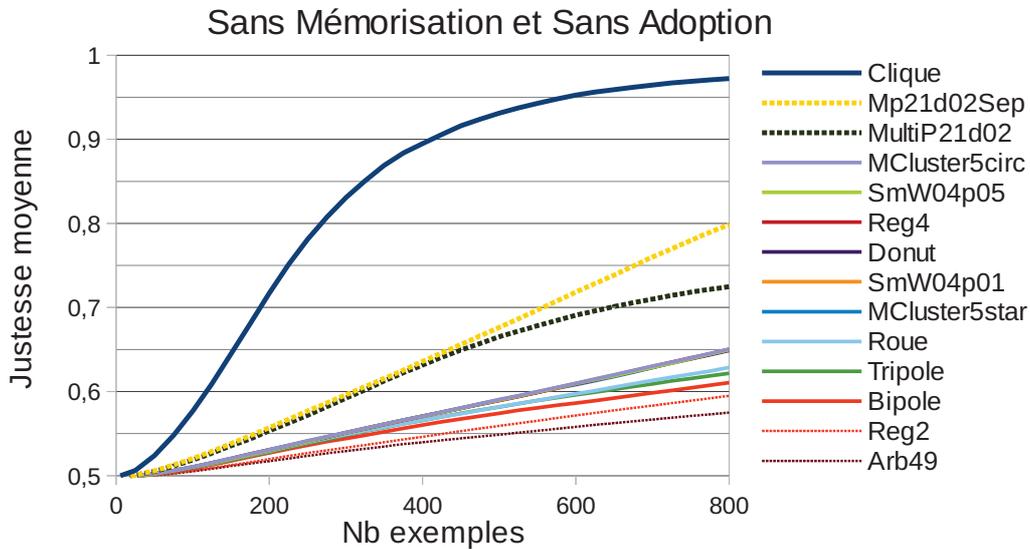


FIGURE 4.9 – Comparaison d’un ensemble de graphes dont l’apprentissage ne comprend pas les mécanismes ni de mémorisation ni d’adoption. La justesse obtenue en fonction du nombre d’exemples acquis par le SMA ne dépend que du degré moyen et de sa répartition.

qu’une ou deux arêtes par sommet secondaire tandis que MultiP21d02sep en a bien plus et l’on peut voir après 400 exemples que le premier graphe est pénalisé par rapport au second. Cet effet s’explique par le fait que les nœuds de plus haut degré apprennent plus vite, mais ne peuvent pas en faire bénéficier leur voisins. Après un certains temps seuls les agents ayant très peu de voisins apprennent encore et la progression est plus lente.

Cas de mémorisation seule

On observe l’évolution de la justesse pour le même ensemble de graphes en présence du mécanisme de mémorisation, mais sans adoption sur la figure 4.10. Du fait de la propagation lente, mais sûre des exemples en présence de mémorisation on voit apparaître des écarts entre les graphes de distance moyenne différentes. On constate toujours que les graphes dont le degré est inégalement réparti ralentissent plus tôt (courbes en pointillé).

Cas d’adoption seule

On observe l’évolution de la justesse pour le même ensemble de graphes en présence du mécanisme d’adoption, mais sans mémorisation sur la figure

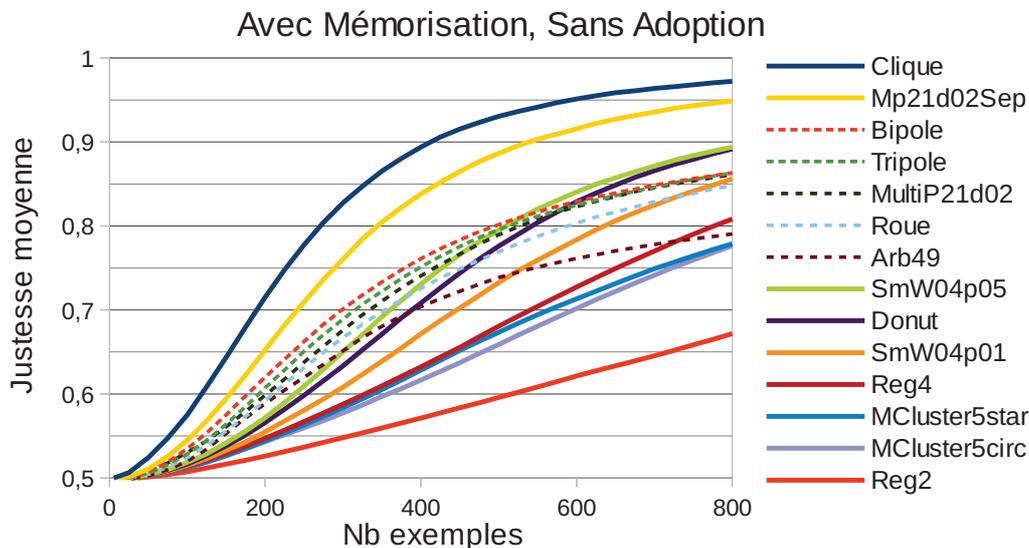


FIGURE 4.10 – Comparaison d’un ensemble de graphes dont l’apprentissage comprend le mécanisme de mémorisation, mais pas celui d’adoption.

4.11. Comme indiqué précédemment on distingue un groupe de courbes correspondant à un apprentissage très rapide pour les graphes de diamètre 2. On remarque également que le Multip21d02 de diamètre 3 bénéficie plus de l’adoption que Multip21d02sep de diamètre 4. Par ailleurs, on peut également remarquer que l’adoption diminue fortement l’effet de la répartition inégale du degré sans pour autant l’effacer complètement. L’apprentissage des graphes de diamètre conséquent ne dépend alors que du degré moyen.

Cas de mémorisation et adoption

On observe l’évolution de la justesse pour le même ensemble de graphes en présence du mécanisme d’adoption et de mémorisation sur la figure 4.12. On remarque parmi les graphes les plus efficaces ceux de diamètre 2. Pour les diamètres plus grand, le degré moyen définit l’efficacité initiale du graphe, mais on observe rapidement l’influence de la distance moyenne, de la répartition inégale des degrés, et du nombre de chemins minimaux de longueur 2.

Pour les graphes de diamètre 2, les SMA qu’ils représentent n’ont presque aucun intérêt dans le processus de mémorisation des exemples externes tant qu’ils bénéficient de l’adoption d’hypothèse. Dès que le diamètre est plus grand, l’adoption n’a plus cette influence prédominante et bien qu’elle soit bénéfique dans tous les cas, la présence de mémorisation fait une plus grande différence dans la qualité de l’apprentissage.

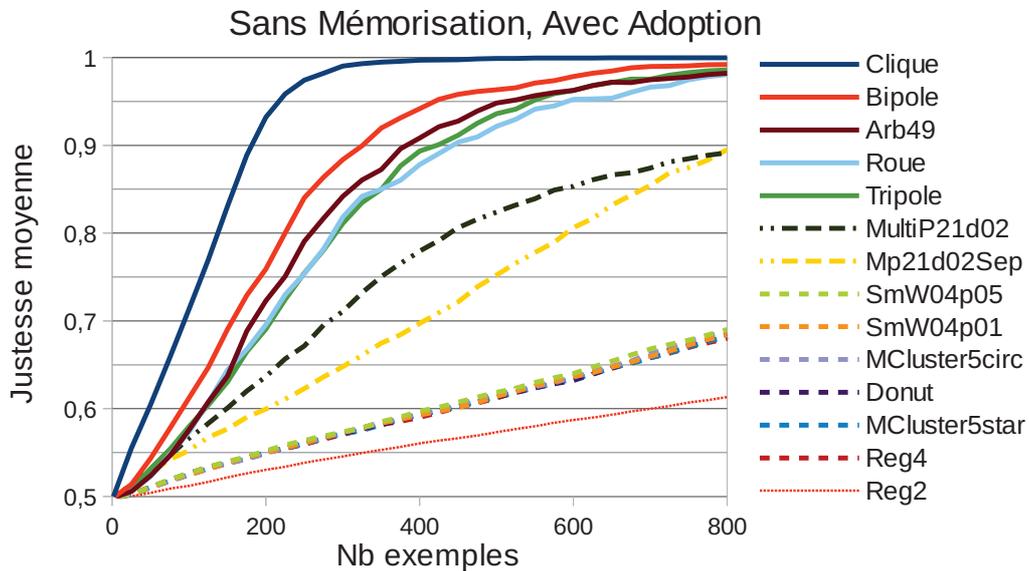


FIGURE 4.11 – Comparaison d’un ensemble de graphes dont l’apprentissage comprend le mécanisme d’adoption, mais pas celui de mémorisation.

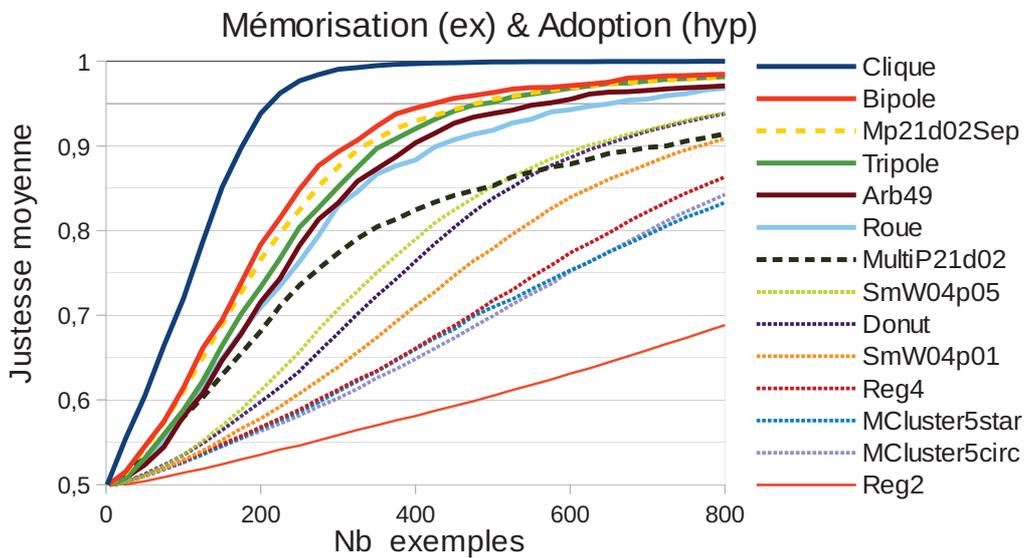


FIGURE 4.12 – Comparaison d’un ensemble de graphes dont l’apprentissage comprend les mécanismes de mémorisation et d’adoption.

Chapitre 5

Apprentissage réseau SMA-consistant : envisager une approche parallèle

L'apprentissage réseau avec des agents autonomes sous-entend l'existence de deux points de vue distincts : au niveau du réseau et au niveau d'un agent. On cherche ainsi à obtenir des propriétés globales de consistance qui reposent sur un comportement individuel et local bien que commun de chaque agent. Nous avons présenté en partie 3.2.2 une première approche de protocole réseau Relay qui se ramène au cas d'une clique par relai puis son amélioration plus dynamique le protocole Variable-Root qui permet d'échanger le rôle d'apprenti pendant une révision globale. Nous verrons ici que ces protocoles sont limités à une exécution séquentielle et comment introduire une approche en parallèle dans les protocoles réseaux. Une approche en parallèle sera proposée au chapitre 6 avec le protocole Waves.

5.1 Pourquoi apprendre en parallèle

Tandis que l'on envisage un plus grand nombre d'agents, on peut être rapidement frustré par la séquentialité du protocole. Pourquoi attendre la réponse d'un agent pour demander au suivant alors qu'ils sont indépendants ? En demandant à tous les voisins en même temps, on ne sera limité que par le temps de réponse du plus lent plutôt que par la somme des temps de réponse de chacun.

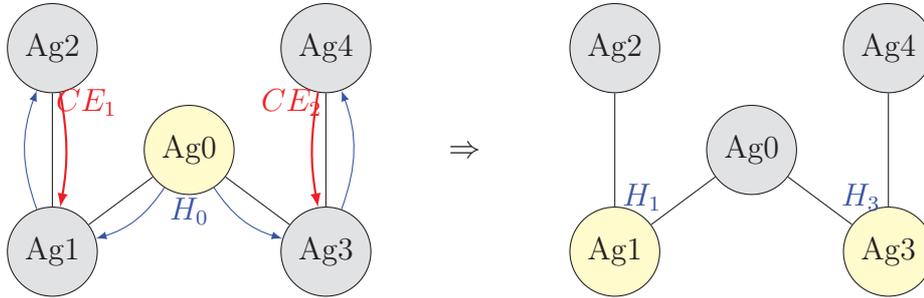


FIGURE 5.1 – L'évolution en parallèle d'une hypothèse H_0 peut, dans le cas de deux contre-exemples différents CE_1 et CE_2 , conduire à la formation de deux hypothèses différentes H_1 et H_3 proposées en même temps par deux apprentis différents $Ag1$ et $Ag3$.

5.2 Apparition de multiples hypothèses

5.2.1 Utilisation du broadcast en réseau avec le protocole Variable-Root

En donnant la possibilité au protocole Variable-root d'effectuer un broadcast c'est-à-dire un envoi simultané de propose à tous ses voisins, le protocole perd toutes ses garanties de terminaison et cohérence. En effet, le fonctionnement en parallèle peut conduire à l'apparition de plusieurs hypothèses comme dans l'exemple 9.

Exemple 9 Prenons un graphe de cinq agents alignés (représenté figure 5.1) où l'agent central $Ag0$ reçoit un exemple donnant lieu à une révision de son hypothèse en H_0 . Lorsque $Ag0$ propose son hypothèse en broadcast, $Ag1$ et $Ag3$ lui répondent chacun par un weak-accept. $Ag0$ leur envoie alors un propagate. De son côté $Ag1$ va contacter $Ag2$ qui lui renvoie un premier contre-exemple CE_1 , tandis que $Ag3$ va contacter $Ag4$ et recevoir un autre contre-exemple CE_2 . $Ag1$ révisé alors H_0 en H_1 et $Ag3$ révisé H_0 en H_3 . $Ag1$ comme $Ag3$ se considèrent alors comme l'apprenti et veulent proposer chacun leur hypothèse à l'ensemble du graphe. Il existe alors deux apprentis distincts et deux hypothèses distinctes.

5.2.2 Problème des exemples multiples

L'arrivée simultanée de plusieurs exemples crée une situation similaire. En effet, en considérant que chacun des exemples reçus soit un contre-exemple

pour l'agent qui le reçoit, ils vont tous déclencher une révision. Le SMA contiendra alors plusieurs apprentis et plusieurs hypothèses en cours de révision globale en même temps. Bien que l'accent soit mis sur la résolution du problème de la parallélisation des communications, le problème des exemples multiples est également traité.

5.3 Conflits entre hypothèses

La rencontre de plusieurs hypothèses peut créer des situations conflictuelles comme l'on peut voir dans l'exemple 10.

Exemple 10 On part d'une situation représentée figure 5.2 où deux agents voisins reçoivent un **propose** simultanément auquel ils répondent par un **weak-accept**. Cela est possible par exemple en partant de la fin de l'exemple 9 au cours duquel *Ag1* a communiqué et révisé plus rapidement, il peut avoir proposé H_1 à *Ag0* qui l'accepte et l'adopte avant que *Ag3* ai fini de former H_3 . *Ag1* demandant à *Ag0* de propager, il propose alors H_1 à *Ag3* pendant que *Ag3* propose H_3 à *Ag0*. On obtient le cas représenté figure 5.2 où deux agents voisins reçoivent un **propose** simultanément auquel ils répondent par un **weak-accept**. Cette situation les amène à changer de père ainsi que d'hypothèse. Ils sont dans la situation où ils attendent un **propagate** pour pouvoir proposer leur nouvelle hypothèse, mais ils ne reçoivent qu'un **weak-accept** de leur père. Les deux agents n'ont pas de réponse prévue à ce **weak-accept** qu'ils ignorent tout en restant en attente l'un de l'autre et le protocole ne peut pas continuer.

5.4 Dépasser la limite du séquentiel

L'exemple d'inter-blocage du protocole Variable-Root lorsqu'on le place dans une situation avec broadcast (exemple 10) est une situation symétrique. Comment s'adapter au problème de la proposition simultanée de deux hypothèses dans un graphe ?

Il y a plusieurs approches envisageables, soit briser la symétrie de la situation pour privilégier une hypothèse et éviter l'inter-blocage soit permettre aux agents de traiter plusieurs hypothèses indépendamment.

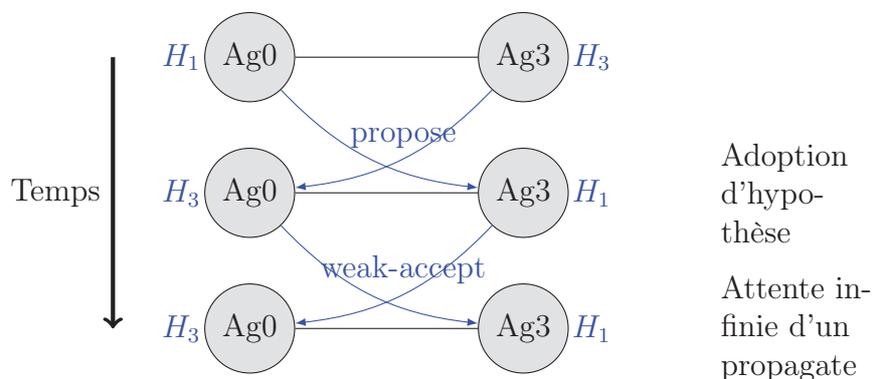


FIGURE 5.2 – Inter-blocage possible du protocole Variable-Root avec broadcast

5.4.1 Choisir immédiatement une des hypothèses

Il faut choisir un critère pour favoriser une hypothèse plutôt qu'une autre lors d'une proposition croisée (lorsque deux agents voisins se proposent deux hypothèses différentes réciproquement et simultanément).

- /!\ En fixant un ordre de priorité sur les agents. Compte tenu de la mentalité de l'apprentissage collaboratif entre agents autonomes qui se veut non hiérarchisée, cette solution est écartée.
- /!\ En fixant une priorité sur le temps (instant de l'envoi) Ce critère n'a pas de sens particulier et permettrait juste de faire disparaître une hypothèse. C'est un critère peu satisfaisant si l'on considère un graphe cyclique et la possibilité que deux hypothèses H_1 et H_2 se croisent en deux endroits différents du graphe et qu'elles soient départagées par un critère local. Il se peut que H_1 prédomine d'un côté et H_2 en l'autre point de rencontre puis qu'elles parcourent le graphe en une boucle infinie. Ce critère ainsi que toute décision purement locale faisant disparaître une hypothèse sont rejetées.
- |?| En fixant, pour les hypothèses, un ordre global que tous les agents connaissent, on peut envisager une adaptation du protocole qui, en cas de conflit, favorise une hypothèse plutôt qu'une autre. Cet ordre global ne servirait alors qu'à maintenir une seule hypothèse à la fois pour qu'elle puisse parcourir l'intégralité du graphe sans risque. Il est possible que l'ordre global fasse disparaître une hypothèse SMA-consistante au profit d'une hypothèse H_c qui ne le soit pas, mais H_c au fil de ses

révisions le deviendra comme dans un protocole séquentiel Variable-Root. Le choix de cet ordre global peut avoir une influence sur la rapidité d'apprentissage (ici en secondes ou en nombre de messages plutôt qu'en nombre d'exemples reçus), mais permettra dans tous les cas à tous les agents d'obtenir une même hypothèse SMA-consistante en fin de révision globale. Bien que ce cas semble résoudre le problème d'interblocage pour un protocole réseau ayant des communications en parallèle, fixer un ordre global sur les hypothèse est une contrainte forte. Ce n'est pas la solution retenue. S'assurer de la faisabilité de cette solution et identifier les difficultés de sa mise en place pourra faire l'objet d'une étude complémentaire qui ne sera pas présentée dans ce document.

5.4.2 Maintenir plusieurs hypothèses

Une autre possibilité est de maintenir toutes les hypothèses créées tant qu'elles ne sont pas contredites. Cela implique de doter l'agent d'une gestion des hypothèses qui ne soit pas limitée à sa seule hypothèse personnelle. Cette solution est également une opportunité pour les agents de s'individualiser un peu plus en ayant des critères personnels de choix d'hypothèse. Ce choix personnel ne devra pas impacter le bon fonctionnement global. Tout agent doit donc étudier attentivement les autres hypothèses dans l'éventualité où elles se montreraient plus intéressantes que son hypothèse personnelle, pour lui-même ou un autre agent. Bien qu'un protocole de ce genre favorise plus l'apparition d'hypothèses que leur disparition, on cherchera lors de sa conception à ce qu'il respecte des propriétés typiques. Le protocole devra en particulier assurer sa terminaison et la SMA-consistance des hypothèses retenues.

Le maintien de plusieurs hypothèses nécessite de différencier l'hypothèse personnelle d'un agent de son hypothèse courante.

Définition 5. L'*hypothèse personnelle* d'un agent est l'hypothèse sur laquelle il s'appuie en phase de restitution. L'*hypothèse courante* est l'hypothèse de travail de l'agent et n'a de sens qu'au cours d'une révision de groupe initiée par cet agent.

Chapitre 6

Waves, un Protocole réseau parallèle

L'objectif, dans ce chapitre est de mettre en place un protocole Waves qui maintienne toutes les hypothèses qui apparaissent et permette la révision et la propagation des hypothèses en parallèle. Il est tout d'abord présenté, ainsi que ses propriétés, puis testé expérimentalement avant d'être amélioré.

6.1 Construction et propriétés

Afin de mieux présenter et comprendre les propriétés et le fonctionnement du protocole Waves, on définit un tour.

Définition 6. Un **tour** correspond à l'ensemble des actions et interactions des agents qui composent un SMA entre deux changements d'état de l'ensemble des exemples présents dans le système. Un tour débute à l'entrée d'un ou plusieurs exemples dans le système et se termine lors de la réception suivante d'exemples.

En pratique, on se placera dans le contexte d'un tour par tour où l'on attendra que plus aucun agent n'ait de message à traiter pour finir le tour. Les agents pourront alors de nouveau recevoir des exemples de l'environnement, marquant ainsi le début d'un nouveau tour. Laisser le temps au système de se stabiliser doit permettre à tous les agents d'obtenir une hypothèse SMA-consistante à la fin de chaque tour. Un tour laisse donc l'ensemble des exemples présents dans le système inchangé, mais pas nécessairement leur répartition. Les hypothèses pourront, quant à elles, évoluer en fonction des exemples qui marquent le début du tour. Si ces exemples sont consistants avec toutes les hypothèses personnelles, le tour les laissera inchangées sinon il correspond à une révision globale.

6.1.1 Propriétés du protocole Waves

On définit un certain nombre d'hypothèses de travail dont certaines pourront être remises en question plus tard :

- ce protocole prend place dans un SMA dont le graphe de communication est toujours connexe.
- aucune arête du graphe n'est modifiée pendant l'apprentissage.
- aucun exemple n'est oublié par aucun agent pendant un tour.
- le concept à apprendre est fixe dans le temps.
- les nouveaux exemples reçus par le SMA sont consistants avec le concept à apprendre (apprentissage sans erreurs).
- l'apprentissage est dit réalisable (un agent peut toujours former une hypothèse consistante avec tout ensemble d'exemples qu'il a reçu).

Principe

On propose un mécanisme de propagation basé sur le comportement individuel d'un agent. Un agent, possédant une hypothèse personnelle, qu'il vient de former ou de réviser à partir de ses exemples, en vérifie la *groupe-consistance* (définie en section 3.2.1) avec ses voisins avant de les informer de cette groupe-consistance. Cette méthode, jusqu'ici identique à celle de SMILE utilisant le broadcast dans une clique, diffère, au moment de la réception des messages de groupe-consistance, pour couvrir le reste du réseau. Un agent, qui reçoit l'information qu'une hypothèse est groupe-consistante pour l'un de ses voisins, cherchera, à son tour, à vérifier si elle est groupe-consistante pour lui-même et ses voisins. On appellera *traiter* une hypothèse le fait de se positionner comme apprenti pour une hypothèse (jusqu'à obtenir une version groupe-consistante de celle-ci, qu'elle nécessite des révisions ou non), ce qui implique donc de la mémoriser et de transmettre l'information de sa groupe-consistance à l'ensemble des voisins. S'assurer de la SMA-consistance d'une hypothèse est alors le rôle de tous les agents et l'on considère la propagation du traitement d'une hypothèse en parallèle avec la transmission du rôle d'apprenti comme représenté figure 6.1.

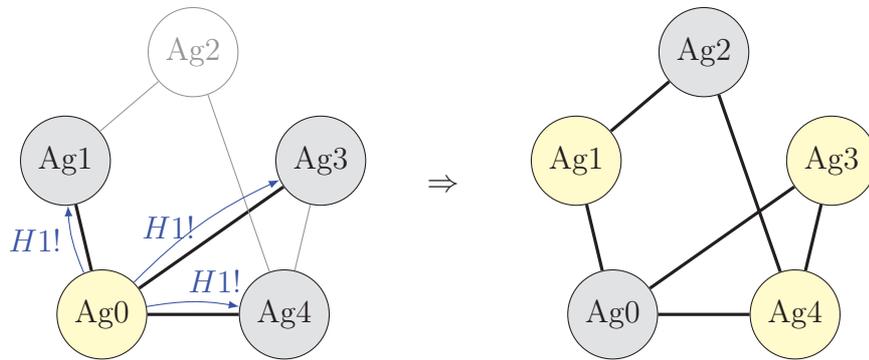


FIGURE 6.1 – Le rôle d’apprenti est repris par les voisins de proche en proche, à la réception d’une hypothèse par sa notification de groupe-consistante.

Propriétés souhaitées

Le protocole Waves doit vérifier plusieurs propriétés :

- la terminaison
- la SMA-consistance de l’hypothèse personnelle de chaque agent en fin de tour.

Afin de mieux comprendre comment ce protocole a été conçu en détails, on propose de chercher à obtenir des propriétés dont l’ensemble est suffisant pour assurer la terminaison et la SMA-consistance recherchées.

Théorème 1.

Dans des conditions de tour par tour sans oubli d’exemples, l’ensemble des propriétés suivantes est suffisant pour qu’un protocole obtienne les propriétés de terminaison et de SMA-consistance de toute hypothèse personnelle en fin de tour.

- p1 Tout nouvel exemple (ou groupe d’exemples) reçu par le système doit être consistant avec les hypothèses personnelles de tous les agents ou donner lieu à une nouvelle hypothèse.
- p2 La réception/mémorisation d’un contre-exemple par un agent conduit cet agent à former d’une nouvelle hypothèse, révisée à partir de celle qui a été contredite et consistante avec tous ses exemples mémorisés.
- p3 Pour chaque formation ou révision d’hypothèse, au moins un agent doit enrichir sa mémoire d’au moins un exemple qu’elle ne contenait pas encore.

- p4 Une hypothèse, formée pendant le tour, est traitée par tout agent de proche en proche dans le graphe jusqu'à ce qu'elle soit contredite par un contre-exemple où que sa consistance soit vérifiée par chaque agent relativement à ses exemples.
- p5 Un agent ne traite pas plusieurs fois la même hypothèse.
- p6 Un agent doit être informé si une hypothèse qu'il a traitée a été contredite et la retirer alors de ses "hypothèses valides".
- p7 Le traitement d'une hypothèse d'un tour plus récent déclenche l'oubli de toute hypothèse d'un tour plus ancien.

Démonstration. On suppose un protocole vérifiant toutes les propriétés du théorème 1.

preuve - partie I (terminaison) : Une nouvelle hypothèse nécessite la mémorisation d'un nouvel exemple pour un agent (*p3*), or il y a un nombre fini e d'exemples présents dans le SMA lors d'un tour et aucun oubli donc il existe un nombre fini d'hypothèses formées pendant un tour. Ce nombre d'hypothèses admet comme borne supérieure $N * e$ dans un réseau connexe de N agents.

Chaque hypothèse est traitée au plus une fois par agent (*p5*) donc on a un nombre fini d'opérations par tour, inférieur à $N^2 * e$, donc le tour est terminal.

preuve - partie II (émergence d'une hypothèse SMA-consistante dès lors que le tour contient une révision) : Une hypothèse formée pendant un tour est, soit SMA-consistante, soit parcourt le graphe jusqu'à être contredite (*p4*) et donne alors naissance à au moins une hypothèse fille (*p2*). Le tour étant terminal (*preuve I*), toute hypothèse formée pendant le tour est, soit SMA-consistante soit a une descendante SMA-consistante.

preuve - partie III (SMA-consistance à la fin d'un tour dès lors qu'il contient une révision) : Un agent mémorise plusieurs hypothèses que l'on peut séparer en trois catégories : les hypothèses d'un tour plus ancien, les hypothèses non SMA-consistantes de ce tour et les hypothèses SMA-consistante de ce tour. La première catégorie n'a pas été confrontée aux nouveaux exemples reçus par le SMA pendant ce tour et n'a donc pas de garantie de SMA-consistance. On souhaite donc qu'un agent ne conserve que la troisième catégorie.

Dès lors qu'une hypothèse est formée pendant le tour, il existe alors une hypothèse SMA-consistante h_{SMA} qui sera formée pendant ce même tour

(*preuve II*). Une hypothèse SMA-consistante ne peut être contredite par aucun exemple d'aucun agent, donc est traitée par tous les agents ($p4$). Le traitement de h_{SMA} entraîne, pour tout agent, l'oubli de toute hypothèse formée pendant un tour plus ancien ($p7$).

Un agent est informé de la contradiction de toute hypothèse qu'il a considérée à tort comme valide ($p6$) et en tient compte. En fin de tour, l'agent ne considère plus comme valide aucune hypothèse de deuxième catégorie (non SMA-consistante). Un agent a eu connaissance de toutes les hypothèses SMA-consistantes formées pendant le tour ($p4$), qu'il compte comme "hypothèses valides".

Un agent compte donc comme "hypothèses valides" en fin de tour toutes les hypothèses SMA-consistantes formées pendant le tour et aucune du tour qui ne soit pas SMA-consistante ni aucune formée lors d'un tour plus ancien. L'hypothèse personnelle d'un agent, étant choisie parmi ses "hypothèses valides", qui contient au moins une hypothèse SMA-consistante (h_{SMA}) et aucune qui ne le soit pas, est SMA-consistante.

preuve - partie IV (SMA-consistance de toute hypothèse personnelle en fin de tour) : D'un tour à l'autre il y a au moins un exemple e_n supplémentaire.

(cas 1) Si e_n (ainsi que les autres nouveaux exemples éventuels) est consistant avec chacune des hypothèses personnelles des agents alors les hypothèses du tour précédent sont toujours SMA-consistantes à ce tour.

(cas 2) Si e_n (ou tout autre nouvel exemple) n'est pas consistant avec une hypothèse alors il y aura formation d'une nouvelle hypothèse pendant ce tour($p1$). Toute hypothèse personnelle de tout agent sera donc SMA-consistante en fin de tour (*preuve III*).

□

Propriétés préalablement vérifiées

Avant de définir plus précisément le protocole Waves, certaines propriétés du théorème 1 sont d'ores et déjà respectées. Le mécanisme de propagation des hypothèses, proposé dans la section précédente, basé sur le traitement d'une hypothèse qu'un voisin a montré groupe-consistante permet d'assurer le comportement de la propriété $p4$. En effet un agent qui forme une hypothèse H_n avec l'aide de la critique de ses voisins, informe ses voisins de sa groupe-consistance. Les voisins réagissant au message de groupe-consistance(H_n) vérifient de nouveau la consistance de l'hypothèse, avec leurs voisins cette fois-ci. Si l'hypothèse n'est toujours pas contredite ils transmettent de nouveau un message de groupe-consistance(H_n) à leurs voisins assurant le traite-

ment de H_n par tout agent de proche en proche. Si au contraire l'hypothèse s'avère non consistante avec les exemples dans la mémoire d'un voisin d'un agent apprenti il la révisé et propose une hypothèse fille au lieu de celle qui est contredite (fonctionnement inchangé de SMILE). On assure ainsi la deuxième partie de la propriété $p4$, mais également la propriété $p2$. La formation d'une nouvelle hypothèse ne peut avoir lieu que suite à la réception d'un exemple contredisant l'hypothèse. Il peut s'agir d'un nouvel exemple provenant de l'environnement ou d'un contre-exemple envoyé par un voisin. Dans ces deux cas l'agent recevant l'exemple le mémorise assurant ainsi la propriété $p3$.

Il faut encore déterminer comment un agent considère une hypothèse reçue par un message de groupe consistante lorsqu'il est déjà en cours de révision d'une autre hypothèse. Afin de respecter l'ensemble des propriétés du théorème 1, il faut également s'assurer qu'une hypothèse n'est pas traitée plus d'une fois ($p5$) par agent et que tout agent est informé lorsqu'une hypothèse qu'il a pu considérer valide est contredite ($p6$). Il faut ajouter une datation de l'hypothèse ($p7$) et potentiellement trouver un moyen pour que chaque agent puisse déterminer en début de tour si un exemple qu'il reçoit ne contredit aucune hypothèse personnelle de n'importe quel agent du SMA entier ($p1$).

6.1.2 Formation du protocole Waves

Pour simplifier le modèle observé, un agent ne traitera qu'une seule hypothèse à la fois. Les hypothèses reçues par message de groupe consistante seront traitées une à une. Elles seront donc mémorisées à leur réception, mais traitées ultérieurement si l'agent est déjà en cours de révision.

Nous verrons dans cette partie comment, organiser la nouvelle mémoire d'hypothèses de l'agent, enrichir le contenu des hypothèses et enfin utiliser ces différentes informations pour former le protocole Waves en vérifiant les propriétés souhaitées.

Mémoire d'hypothèses multiples

Un agent Waves, bien que toujours sur le format d'un agent SMILE, dispose d'une mémoire des hypothèses qui ne se limite plus à une hypothèse personnelle. Un agent dispose, s'il est apprenti, d'une hypothèse courante, l'hypothèse qu'il est en train de traiter. Afin de ne pas traiter plusieurs fois une même hypothèse, un agent conserve également un historique de toutes les hypothèses qu'il a rencontrées au cours du tour courant. Cette mémoire d'hypothèses est appelée "hypothèses à filtrer" pour souligner que, dès lors qu'une hypothèse H_{af} est en mémoire, un agent peut ignorer tout message

de groupe-consistance de H_{af} envoyé par un voisin. C'est un identifiant personnel à l'hypothèse qui permet de la reconnaître, il sera présenté ci-après dans cette partie. Les "hypothèses à filtrer" regroupent trois ensembles d'hypothèses distincts ainsi que l'hypothèse courante comme défini ci-dessous et illustré par la figure 6.2. Une des hypothèses en mémoire est également identifiée comme étant l'hypothèse personnelle de l'agent.

Définition 7. Les *hypothèses à filtrer* (figure 6.2) d'un agent sont l'ensemble des hypothèses que cet agent compte dans sa mémoire. La mémoire d'hypothèses se décompose en trois ensembles d'hypothèses, définis ci-après, plus le singleton de l'hypothèse courante.

La première de ces catégories constitue les *hypothèses à traiter* que l'agent a reçu de ses voisins par messages de groupe-consistance, mais qu'il n'a pas encore traité lui-même.

La deuxième catégorie est l'ensemble des *hypothèses vérifiées*. Ce sont les hypothèses que l'agent a traitées et qu'il a notifiées comme étant groupe-consistante. Elles sont considérées comme "potentiellement SMA-consistantes" et donc candidates pour être l'hypothèse personnelle. Si l'agent apprend qu'une de ces hypothèses est contredite plus tard, il la déplacera vers la dernière catégorie.

La troisième catégorie est constituée des hypothèses qui, de manière certaine, ne sont pas SMA-consistantes. Ce sont les *hypothèses rejetées*.

Un agent garde en mémoire¹ toutes les hypothèses dont il a connaissance. Il filtre les hypothèses qui lui sont notifiées pour n'ajouter aux hypothèses à traiter que les hypothèses qu'il ne connaît pas encore. Les hypothèses à traiter sont traitées successivement une seule fois chacune. La propriété $p5$ est donc vérifiée. On peut donc déjà assurer la terminaison du protocole avec la partie I de la démonstration du théorème 1.

Format d'une hypothèse

On souhaite mettre en place les propriétés $p6$ et $p7$, en enrichissant la structure d'une hypothèse.

Identifiant. On choisit d'identifier toute hypothèse de manière unique afin de pouvoir garder une trace simple de son existence. Il est alors possible de l'identifier à sa réception, mais aussi, de conserver et transmettre l'information de sa contradiction de manière succincte. Une hypothèse rejetée est réduite à son identifiant.

1. L'hypothèse est réduite à un identifiant unique si elle est rejetée et conservée dans son intégralité sinon.

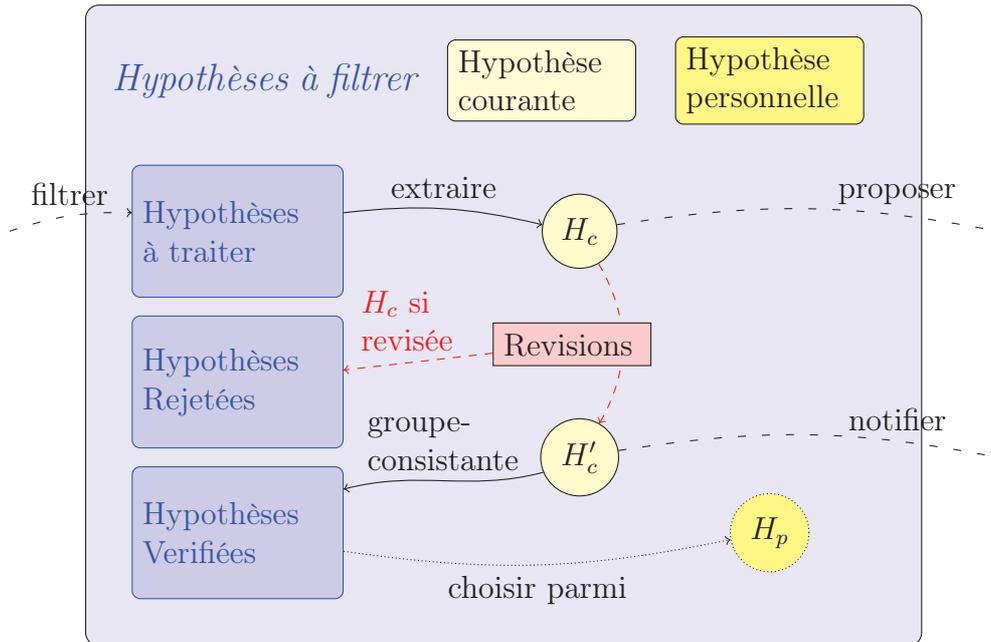


FIGURE 6.2 – L’agent garde en mémoire de nombreuses hypothèses et interdit l’arrivée de toute hypothèse déjà présente. L’hypothèse courante H_c de l’agent est choisie parmi les hypothèses à traiter tant que l’ensemble n’est pas vide. Si H_c n’est pas groupe-consistante, elle est révisée jusqu’à ce qu’une de ses descendantes H'_c le soit. Dans ce cas H_c n’est pas SMA-consistante, elle est alors rangée dans les hypothèses rejetées. Si H_c est groupe-consistante, aucune révision n’est nécessaire, $H'_c = H_c$. Dans tous les cas l’hypothèse groupe-consistante obtenue est ajoutée à l’ensemble des hypothèses vérifiées. L’hypothèse personnelle H_p peut être choisie à tout moment, par l’agent, selon ses critères personnels, parmi ses hypothèses vérifiées. H_p est simplement identifiée, mais pas retirée de l’ensemble des hypothèses vérifiées.

Historique. Une hypothèse H_c qu'un agent a reçu d'un de ses voisins et qu'il doit réviser suite à la réception d'un contre-exemple est une hypothèse qui a été reconnue comme "vérifiée" par le dit voisin et potentiellement d'autres agents. Il faut donc avertir ces agents de la non-SMA-consistance de H_c . On utilise l'hypothèse fille comme vecteur de cette information en lui ajoutant un historique. L'historique d'une hypothèse contient les identifiants de tous ses *ancêtres*, c'est à dire l'identifiant de sa mère et des ancêtres de sa mère.

Définition 8. Un **ancêtre** d'une hypothèse H_f se définit par récurrence comme la mère H_m de H_f où tout ancêtre de H_m . (On considère des hypothèses filles et mères, selon la définition 4.)

Comme vu dans la partie II de la démonstration du théorème 1, une hypothèse est, soit SMA-consistante, soit a au moins une fille et comme le tour est terminal, cette fille ou l'une de ses descendante est SMA-consistante. Une hypothèse SMA-consistante ne peut être contredite donc parcourt l'ensemble du graphe avec son historique. Tous les agents reçoivent donc l'information de la contradiction de toute hypothèse qui a été révisée pendant le tour. La propriété *p6* est ainsi vérifiée.

Date. Une hypothèse parcourt l'ensemble du graphe à une certaine date. Cette date témoigne d'un état du système en matière d'exemples présents. Une hypothèse qui parcourt le graphe à une date plus récente tient donc compte d'exemples supplémentaires. On dote les hypothèses d'un champ *date*, fixé à sa création comme la date la plus récente entre la date de la mère et celles des exemples en mémoire de l'agent. Tout exemple reçu avant la date de cette hypothèse, sera confronté à l'hypothèse. Une hypothèse plus récente indique l'existence d'exemples qui ne sont pas garantis comme consistants avec toute hypothèse plus ancienne.

Les hypothèses plus anciennes n'ayant plus de garanties, on choisit de les rejeter. Pour alléger la mémoire des agents et la gestion des hypothèses, les hypothèses plus anciennes sont intégralement oubliées. Les agents ont cependant toujours au moins une hypothèse en mémoire puisque cet oubli ne peut être déclenché que par l'arrivée d'une nouvelle hypothèse.

Cette gestion de la date permet de garantir la propriété *p7*.

Pré-traitement

L'hypothèse du format SMILE est encapsulée dans une structure qui dispose de trois nouvelles informations : *la date*, *un identifiant* et un ensemble d'ancêtres de même date, qui constitue *l'historique* des révisions de cette

Définition 9. Une **hypothèse Waves** est formée par en-capsulation d’une hypothèse SMILE de référence. Elle est constituée de :

- un identifiant (couple d’entiers unique)
- un historique (ensemble d’identifiants correspondant aux ancêtres)
- une date (date de création comptée en nombre de tours)
- un contenu logique (le contenu d’une hypothèse SMILE)

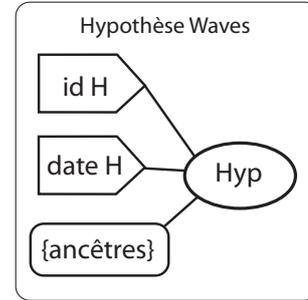


FIGURE 6.3 – Structure d’une hypothèse Waves

hypothèse pendant un tour. Afin de tenir compte de toutes les informations ajoutées aux hypothèses d’une part et à la mémoire d’hypothèses des agents d’autre part, il faut mettre en place un *pré-traitement* défini ci-après. Toute hypothèse qui arrive dans un message de groupe-consistance est pré-traitée avant d’être ajoutée aux hypothèses à traiter ou de devenir l’hypothèse actuelle.

Définition 10. Pré-traitement : action du protocole d’un agent a_r à la réception d’un message de groupe-consistance d’une hypothèse H_g . Il permet de déterminer s’il doit traiter H_g puis sa priorité de stockage. Si H_g est retenue, le pré-traitement met également à jour la mémoire d’hypothèses de a_r en fonction des informations apportées par H_g telles que la date ou les ancêtres.

Le pré-traitement permet d’une part de ne traiter qu’une seule fois au plus chaque hypothèse et d’autre part de mettre à jour la mémoire des hypothèses de l’agent avec les informations de l’hypothèse pré-traitée. Il consiste en trois étapes de filtre et une étape de stockage comme détaillé dans l’algorithme section 6.1.3 et schématisé figure 6.4.

L’hypothèse personnelle n’est pas garantie SMA-consistante avant la fin d’un tour, mais peut être choisie parmi les hypothèses vérifiées à tout moment en cas de restitution précoce. Ce choix peut être fait selon des critères personnels qui restent à définir et étudier. Le choix trivial retenu pour les premiers tests est de prendre pour hypothèse personnelle la dernière hypothèse courante de l’agent parmi les hypothèses vérifiées.

En fin de tour l’ensemble des hypothèses vérifiées est commun à tous les agents et les agents choisissent leur hypothèse personnelle parmi celles-ci. Les

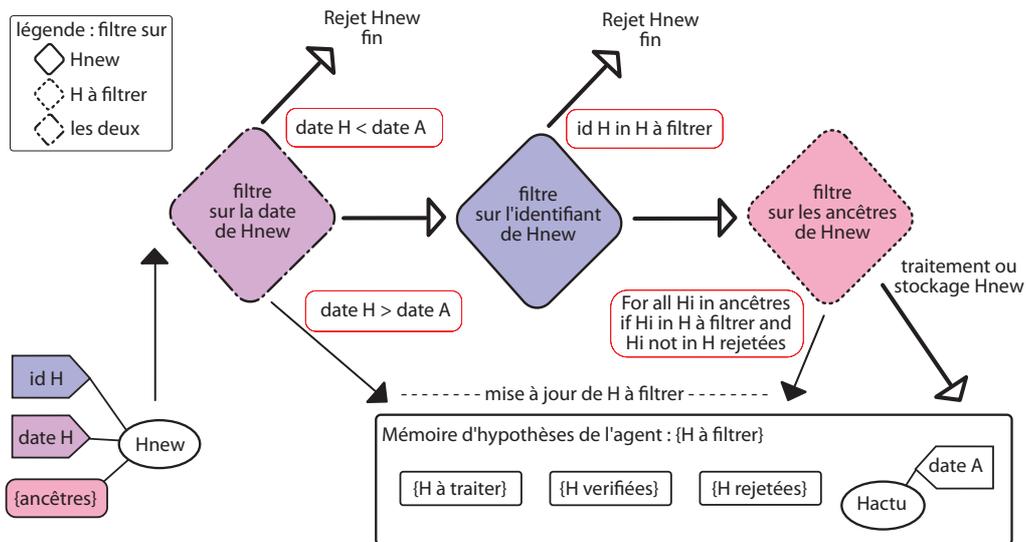


FIGURE 6.4 – Pré-traitement d’une hypothèse $Hnew$ reçue par message de groupe-consistance d’un voisin. Les critères de date puis d’identifiant doivent être respectés par $Hnew$ pour qu’elle ne soit pas simplement ignorée par l’agent. La date ainsi que les ancêtres de $Hnew$ permettent de mettre à jour la mémoire d’hypothèses de l’agent en rejetant les hypothèses obsolètes. Si $Hnew$ est conservée, elle est ajoutée aux hypothèses à traiter ou devient immédiatement l’hypothèse courante, selon que l’agent était en cours de révision ou au repos.

hypothèses vérifiées ne contiennent que des hypothèses SMA-consistantes et une hypothèse SMA-consistante est connue de tous. En respectant le tour par tour, on peut donc se contenter de vérifier la consistance des nouveaux exemples par rapport à l'ensemble des hypothèses vérifiées au tour précédent pour respecter la propriété $p1$ comme détaillé dans l'algorithme en section 6.1.3. Il est possible qu'une hypothèse vérifiée, non consistante avec le nouvel exemple, ne soit en réalité pas choisie comme hypothèse personnelle par aucun agent, mais sa révision superflue ne remet pas en cause la preuve du théorème 1.

Les propriétés $p1$ à $p7$ sont toutes vérifiées et sont suffisantes pour obtenir les propriétés souhaitées d'après le théorème 1 donc le protocole Waves termine et tout agent dispose d'une hypothèse personnelle SMA-consistante en fin de tour.

La section suivante présente Waves de manière algorithmique suivie d'une dernière section constituée d'exemples illustratifs de son exécution.

6.1.3 Version algorithmique du protocole Waves

En pratique un protocole correspond à un ensemble de comportements de l'agent à la lecture de chaque message reçu. Un agent exécute l'ensemble des actions relatives à la lecture d'un message avant de passer à la lecture du suivant. Les messages sont reçus dans une file de message en attendant d'être lus. Les messages utilisés par le protocole Waves diffèrent légèrement de ceux du protocole Variable-Root.

Dans le protocole Waves, un agent A peut être dans plusieurs états distincts : *apprentice* correspond au moment où l'agent est en cours de révision et attend des réponses concernant un **propose** tandis que l'état *waiting* est actif lorsqu'il n'a pas d'hypothèse à traiter, ni courante. L'hypothèse personnelle h_p de A est mise à jour à chaque modification des hypothèses vérifiées. A dispose de deux compteurs de messages nA et nR , respectivement le nombre d'**accept** reçus et le nombre de réponses reçues ainsi que d'un buffer de contre-exemples.

Comportement à la réception de :

- **propose**(h) A vérifie la consistance de h avec ses exemples en mémoire et répond par un message **counter-example**(ce) s'il détient un exemple ce qui contredit h ou un message **accept**(h) sinon.
- **accept**(h) Si A est "apprentice" et que l'identifiant de h est l'identifiant de h_c , A incrémente nA et nR de 1. Si nR est égal au nombre de voisins alors selon le cas :

- **cas AAbandonner** (h est marquée "à abandonner")
 A range h dans les hypothèses rejetées et commence à traiter l'hypothèse à traiter suivante.
 - **cas Ac1** (h n'est pas marquée "à abandonner" et nA est égal au nombre de voisins) :
 - * Si l'identifiant d'origine (voir traitement ci-après) est non vide, il est ajouté aux ancêtres de h_c .
 - * A envoie un message `group-consistant(h_c)`
 - * h_c est mémorisée dans les hypothèses vérifiées.
 - * Selon l'état de l'ensemble des hypothèses à traiter :
 - Si (HaT) est non-vide, A commence le traitement de l'hypothèse suivante.
 - Si HaT est vide A passe de l'état "apprenti" à celui de "waiting".
 - **cas Ac2** (h n'est pas marquée "à abandonner" et nA n'est pas égal au nombre de voisins) :
 - * A révisé h_c en tenant compte des contre-exemples reçus dans le buffer, qu'il mémorise avant de vider le buffer.
 - * h_c reçoit un nouvel identifiant et son identifiant d'origine est ajouté aux hypothèses rejetées s'il n'y est pas déjà.
 - * Les compteurs de messages sont ré-initialisés et A propose la nouvelle h_c .
- `counter-example(ce)` Si A est "apprenti" et que ce est bien un contre-exemple de (h_c) alors A incrémente nR de 1 et stock ce dans le buffer de contre-exemples. Si nR est égal au nombre de voisins, alors l'agent se comporte selon la circonstance comme dans le cas AAbandonner ou le cas Ac2.
 - `group-consistant(h)` A déclenche un pré-traitement de l'hypothèse h (cf ci-dessous).
 - `new-example(e)` A regroupe les nouveaux exemples en attendant le début du tour.
 - `begin-turn` A vérifie la consistance de h_p puis celle de toutes ses hypothèses vérifiées avec l'ensemble des nouveaux exemples jusqu'à ce qu'elles soient toutes consistantes ou que l'une d'elle soit contredite. Si

une hypothèse h_{ctr} est contredite, A oublie toute sa mémoire d'hypothèses à l'exception de h_{ctr} . h_{ctr} efface également son historique d'ancêtres et prend la date actuelle (celle des nouveaux exemples), puis A révisé h_{ctr} et en fait le traitement.

Le *pré-traitement* d'une hypothèse h par un agent A se fait en quatre étapes :

- vérifier la date de h . Si la date est plus ancienne que celle de h_c , l'agent passe au message suivant en ignorant h . Si la date est plus récente que les dates des hypothèses de A , A efface toutes ses hypothèses en mémoire et commence le traitement de h (voir ci dessous) après ajout de ses ancêtres aux hypothèses rejetées. Si la date de h est égale à celle de h_c , A passe à l'étape suivante.
- vérifier si h fait partie des hypothèses à filtrer. Si oui l'agent passe au message suivant sinon on continue.
- tenir compte des ancêtres de h . Tous les ancêtres de h sont ajoutés aux hypothèses rejetées de A . Si un de ces ancêtres était présent dans les hypothèses à traiter ou vérifiées, il en est retiré. Si l'hypothèse courante fait partie des ancêtres de h , A la marque "à abandonner".
- Si HaT est vide et A est dans l'état "waiting", A commence à traiter h sinon h est ajoutée aux hypothèses à traiter.

Le *traitement* d'une hypothèse h par un agent A commence par faire de h son hypothèse courante h_c et remettre les compteurs de messages (nR et nA) à zero. Si A n'était pas "apprentice", il le devient. A enregistre un *identifiant d'origine* qui est soit vide dans le cas d'une hypothèse formée à partir de nouveaux exemples soit l'identifiant de h si elle a été reçue par message de groupe-consistance. A propose h_c en broadcast.

6.1.4 Exemples d'états mémoire d'agents Waves lors d'un tour

En prenant le graphe composé de 4 agents (A, B, C, D) dont les communications sont toutes possibles sauf entre A et D et afin d'illustrer diverses situations qui modifient la mémoire des hypothèses, nous considérons un tour (illustré figure 6.5 à 6.8) se déroulant de la manière suivante :

- I A reçoit un exemple e_1 qui est consistant avec son hypothèse personnelle h_{a1} , mais pas consistant avec une autre de ses hypothèses Vérifiées, h_{a2} . A révisé h_{a2} en h_2 et la propose à B et C qui renvoient un accept.
 - II A envoie un message de groupe-consistance(h_2) à B et C qui proposent h_2 à tous leurs voisins.
 - III D envoie un contre-exemple ce à B (puis à C) qui révisé h_2 en h_{2b} dont la proposition est acceptée par tous. De même C révisé h_2 en h_{2c} qui peut être différente de h_{2b} .
 - IV B envoie un message de groupe-consistance(h_{2b}) en broadcast avant que C n'ai reçu les réponses à ses propositions de h_{2c} .
 - V h_{2c} est acceptée par tous et C envoie un message de groupe-consistance (h_{2c}) avant de traiter h_{2b} . A et D traitent h_{2b} puis h_{2c} .
- fin Le tour se termine avec A, B et D ayant h_{2c} comme hypothèse personnelle tandis que C a h_{2b} .

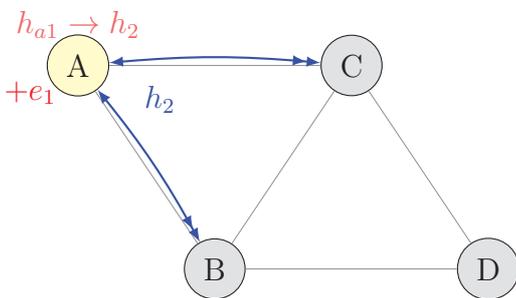


FIGURE 6.5 – Protocole Waves (exemple de tour, étapes I et II). La nouvelle hypothèse h_2 est transmise à B et C suite à un échange de messages (propose, accept, groupe-consistance)

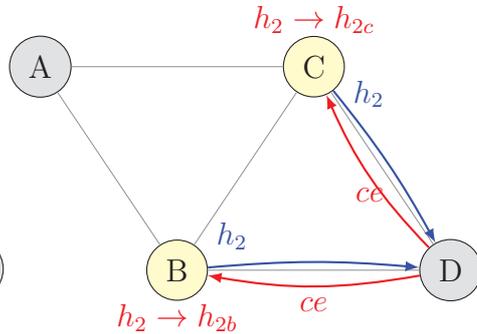


FIGURE 6.6 – Protocole Waves (exemple de tour, étape III). B et C reçoivent de D un contre-exemple ce en réponse à leur proposition de h_2 qu'ils révisent en conséquence.

Cas de la réception d'un nouvel exemple :

(étape I) La consistance de l'exemple e_1 est testée en priorité avec l'hypothèse personnelle de A qui le reçoit puis avec toutes les hypothèses Vérifiées jusqu'à ce qu'une hypothèse soit inconsistante avec e_1 , ici h_{a2} ou qu'elles

soient toutes consistantes. La révision de ha_2 conduit à la formation de h_2 qui est plus récente et déclenche l'oubli de toutes les autres hypothèses de A .

$$\text{Mémoire de } A : \begin{cases} h_{actuelle} : h_{a1} & \rightarrow h_2 \\ A \text{ traiter} : \emptyset & \rightarrow \emptyset \\ V\acute{e}rifi\acute{e}es : \{h_{a1}, h_{a2}\} & \rightarrow \emptyset \\ Rejet\acute{e}es : \{\dots\} & \rightarrow \emptyset \end{cases}$$

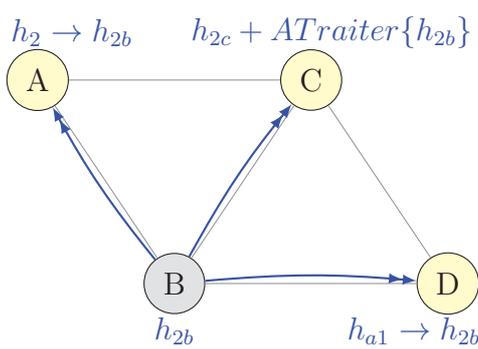


FIGURE 6.7 – Protocole Waves (exemple de tour, étape IV). B envoie un message de groupe-consistance à A , C et D qui adaptent leur mémoire d'hypothèse.

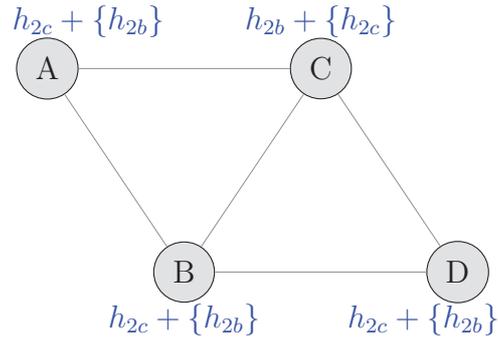


FIGURE 6.8 – Protocole Waves (exemple de tour, fin de tour). A , B et D traitent h_{2c} en dernier tandis que C traite h_{2b} en dernier, mais leur ensemble d'hypothèses vérifiées est le même.

Cas de la réception d'un message de groupe-consistance :

(étape IV, figure 6.7) Les trois agents, A , C et D , qui reçoivent le message de groupe-consistance de h_{2b} , envoyé par B , sont dans des situations différentes.

D n'a pas encore traité d'hypothèse de ce tour, il doit d'abord oublier ses anciennes hypothèses avant d'apprendre les informations apportées par h_{2b} et la traiter. Il mémorise notamment son ancêtre h_2 .

$$\text{Mémoire de } D : \begin{cases} h_{actuelle} : h_{a1} & \rightarrow h_{2b}(\text{hist} : \{h_2\}) \\ A \text{ traiter} : \emptyset & \rightarrow \emptyset \\ V\acute{e}rifi\acute{e}es : \{h_{a1}, h_{a2}\} & \rightarrow \emptyset \\ Rejet\acute{e}es : \{\dots\} & \rightarrow \{h_2\} \end{cases}$$

A a déjà traité une hypothèse pendant ce tour, il vérifie tout d'abord que h_{2b} ne fait pas partie des hypothèses à filtrer (n'est pas déjà en mémoire) puis il doit alors confronter ses hypothèses Vérifiées à l'historique de h_{2b} . h_2 est retirée des hypothèses Vérifiées et ajoutée aux Rejetées.

$$\text{Mémoire de } A : \begin{cases} h_{actuelle} : h_2(hist : \emptyset) & \rightarrow h_{2b}(hist : \{h_2\}) \\ A \text{ traiter} : \emptyset & \rightarrow \emptyset \\ V\acute{e}rifi\acute{e}es : \{h_2\} & \rightarrow \emptyset \\ Rejet\acute{e}es : \emptyset & \rightarrow \{h_2\} \end{cases}$$

C est actuellement apprenti, il doit finir sa r\evision en cours avant de pouvoir traiter h_{2b} nouvellement re\c{u}e. h_{2b} est donc stock\ee temporairement dans les hypoth\eses \aa traiter.

$$\text{Mémoire de } C : \begin{cases} h_{actuelle} : h_{2c}(hist : \{h_2\}) & \rightarrow h_{2c}(hist : \{h_2\}) \\ A \text{ traiter} : \emptyset & \rightarrow h_{2b} \\ V\acute{e}rifi\acute{e}es : \emptyset & \rightarrow \emptyset \\ Rejet\acute{e}es : \{h_2\} & \rightarrow \{h_2\} \end{cases}$$

Cas de la r\ec{e}ption d'une hypoth\ese d\ej\aa connue :

Lorsque les derniers messages de groupe-consistance sont re\c{u}s, ils concernent des hypoth\eses \aa filtrer et sont donc ignor\es. Par exemple \aa la fin de l'\etape V, B envoie un message de groupe-consistance (h_{2c}) \aa C . h_{2c} est V\erifi\ee pour C donc \aa fortiori c'est une hypoth\ese \aa filtrer, elle n'est donc pas trait\ee de nouveau.

Fin de tour :

Remarque : \aa part l'hypoth\ese actuelle (et personnelle), la m\emoire d'hypoth\eses des agents est la m\eme en fin de tour.

$$\text{Mémoire de } (A,B,C,D) : \begin{cases} A \text{ traiter} : \emptyset \\ V\acute{e}rifi\acute{e}es : \{h_{2b}, h_{2c}\} \\ Rejet\acute{e}es : \{h_2\} \end{cases}$$

6.2 R\esultats exp\erimentaux

Une exp\erience est compos\ee d'une centaine d'ex\ecutions diff\erentes, chacune correspondant \aa une s\eqvence d'exemples acquis incr\ementalement par les agents choisis al\eaatoirement dans le r\eseau. Les exp\erience suivantes sont r\eealis\ees sur une machine de 192 Go de RAM avec 48 c\oeurs de 2.60 Ghz de mod\ele Intel Xeon 2690. Pour s'assurer que chaque agent puisse avoir l'exclusivit\ee des ressources d'un c\oeur et donc b\en\eficier du fonctionnement en parall\ele, on s'int\eresse particuli\erement \aa des groupes de 40 agents. Les n\oeuds suppl\ementaires sont utilis\es pour la simulation de l'environnement et la r\ecolte des r\esultats.

6.2.1 Paramètres

Protocole. Pour situer les résultats de Waves, on utilise trois protocoles de référence. L'apprentissage d'un agent seul permet de situer l'efficacité du processus d'apprentissage (Imind2) indépendamment de tout effet multi-agents. SMILE séquentiel avec mémorisation et adoption permet de montrer l'effet multi-agents en Clique, il peut améliorer la vitesse d'apprentissage pour des problèmes difficiles. Enfin le protocole Variable-Root est la référence qui s'exécute dans un réseau. Waves est le seul à pouvoir à la fois avoir de multiples apprentis et fonctionner dans un réseau donc recevoir de multiples exemples et propager en parallèle dans un réseau.

Problème d'apprentissage. Nous choisissons d'utiliser principalement le multiplexer-11 (M11) présenté en 2.1.3. Étant un problème booléen difficile, SMILE bénéficie d'une meilleure exploration multi-agents que l'on souhaite retrouver avec Waves.

Graphe de communications. Afin de couvrir des structures de graphe et des propriétés variés, nous choisissons d'étudier 6 formes de graphe différentes, toutes connexes. La Clique, l'Arbre5, la Roue, un Small-World smW4p05, la Ligne et le Cercle. La Ligne et le Cercle représentent les pire cas, tandis que Arbre5 qui peut représenter une structure hiérarchique présente des distances courtes entre paires d'agents et aucun cycle. La Roue contient des distances, mais aussi des cycles, très courts ainsi qu'un nœud de degré maximal. La Clique est un cas extrême sans contraintes de communication. C'est la seule structure possible pour SMILE. Enfin le Small-World représente une structure fréquente. C'est aussi un cas équilibré avec des distances variées à tendance courtes ainsi que de nombreux cycles.

Choix de l'hypothèse personnelle. Une des particularité de Waves est de maintenir plusieurs hypothèses et de permettre aux agents de choisir parmi celles qu'il considère Vérifiées. On propose de comparer trois choix possibles bien que d'autres puissent être envisagés. Le choix par défaut est la dernière hypothèse qui ait été hypothèse courante (LastSeen). On le compare à un choix aléatoire (Random) et l'on considère un critère de qualité sur l'hypothèse, la taille (définie en section 2.1.3) la plus courte (Shorter).

Nombre d'exemples par tour. Il est possible avec Waves de considérer l'acquisition de plusieurs exemples simultanés répartis sur divers agents du graphe. On montrera en particulier l'influence d'un rythme à 10EPT (exemples par tour) par rapport à celui de 1EPT par défaut.

Priorité des Hypothèses à traiter. On considère deux options concernant l'ordre dans lequel extraire la prochaine hypothèse courante de l'ensemble des hypothèses à traiter. La première option est de considérer l'ensemble comme une file avec une priorité FIFO (First in first out) qui consiste à choisir en premier l'hypothèse présente depuis plus longtemps. La seconde option considère l'ensemble comme une pile LIFO (Last In First Out) qui extrait en priorité l'hypothèse la plus récemment reçue.

6.2.2 Résultats et analyse

Option LIFO strictement supérieure à FIFO

Une étude préliminaire concernant la priorité des Hypothèses à traiter montre sans ambiguïté que l'option LIFO est préférable à la FIFO. En effet comme le montre le tableau 6.1, cette option n'a aucun effet sur la justesse des hypothèses personnelles des agents, mais peut avoir un gros impact sur le temps d'exécution. Cette différence n'est visible que sur des graphes contenant des cycles où une hypothèse fille peut être reçue avant que sa mère ne soit traitée. Un agent peut diminuer le nombre d'hypothèses qu'il a à traiter et donc le nombre d'opérations élémentaires et le temps d'exécution. Ce cas favorable est d'autant plus fréquent quand le traitement est effectué en LIFO.

TABLE 6.1 – Influence de la priorité d'extraction des "Hypothèses à traiter"

Waves	Tree5				SmW4p05			
	1 EPT		10 EPT		1 EPT		10 EPT	
	FIFO	LIFO	FIFO	LIFO	FIFO	LIFO	FIFO	LIFO
justesse 300ex	0,994	0.992	0,983	0,980	0.976	0.977	0.959	0.96
temps (ms)	452	445	662	673	4921	2829	9826	4768

L'option LIFO est donc l'option par défaut de priorité des hypothèses à traiter de Waves.

Les caractéristiques des graphes révèlent les forces et faiblesses de Waves

On limite tout d'abord l'étude à 1EPT. La forme du graphe de communication semble avoir plus d'influence sur Waves qu'elle n'en a sur Variable-Root² tant en termes de justesse que de couts (temps d'exécution, nombre de messages, nombre de subsomptions). Waves présente néanmoins dans tous les

2. on utilise le protocole Variable-Root qui construit un arbre couvrant en profondeur d'abord. Il existe une autre version de construction en largeur d'abord. Elle diminue ses

cas une précision au moins égale à celle de l'agent seul. Dans le cas du M11, il présente également, comme SMILE et Variable-Root, un gain exploratoire de justesse, effet d'une meilleure exploration de l'espace des hypothèses par un système multi-agents.

Comme l'on peut voir sur la figure 6.9, une distance moyenne entre paires d'agents grande, comme pour Ligne et Cercle, signifie beaucoup de messages et d'intermédiaires donc un temps d'exécution conséquent.

On analyse les résultats de Waves des figures 6.9 et 6.10. Waves bénéficie naturellement des nœuds de haut degré comme dans Arbre5 et Roue tandis que la présence de cycles ralentit l'apprentissage. Les cycles sont d'autant plus coûteux qu'ils sont longs et nombreux. Ceux de la Roue, très courts, ont peu d'influence tandis que le Small-World est affecté par leur présence. De même le Cercle nécessite plus de temps, de messages et de calculs, que la Ligne pour compléter son apprentissage. Une longue distance, d'autant plus à l'intérieur d'un cycle, augmente la probabilité d'avoir plusieurs hypothèses étudiées en même temps dans le graphe et donc plus de travail d'apprentissage pour les agents. L'augmentation des hypothèses va de pair avec l'augmentation des révisions des agents et donc de la redondance de la mémoire des agents. Les agents ont plus d'exemples en commun en mémoire ce qui diminue la variabilité de l'exploration de l'espace des hypothèses. Waves en présence de cycles perd donc un peu en justesse par rapport à Variable-Root en bénéficiant moins de l'effet exploratoire multi-agents bénéfique dans les problèmes difficiles.

On peut résumer les forces de Waves ainsi que les faiblesses qu'elles impliquent en deux idées principales :

- propagation dynamique : tandis que Variable-Root retire tous les cycles en définissant un arbre couvrant avant de procéder à un apprentissage, Waves propage sur le moment partout où c'est possible. Au lieu d'un parcours selon un arbre conçu de manière statique sur le simple critère de l'existence ou non de lien, les hypothèses se déplacent dans un arbre dynamique qui dépend de la réactivité des agents à chaque instant. Cet avantage de conception a malheureusement un coût élevé en termes de communications entre les branches de l'arbre ainsi parcouru.
- broadcast et apprentis multiples : Les nœuds de haut degrés donnent un

temps d'exécution pour les topologies de Roue (50%) et Clique (20%) bien qu'ils restent supérieurs à ceux de Waves. Il est possible également de diminuer son temps pour SmW4P05 (17%), mais les temps de Cercle (+40%) et Ligne (+50%) sont supérieurs.

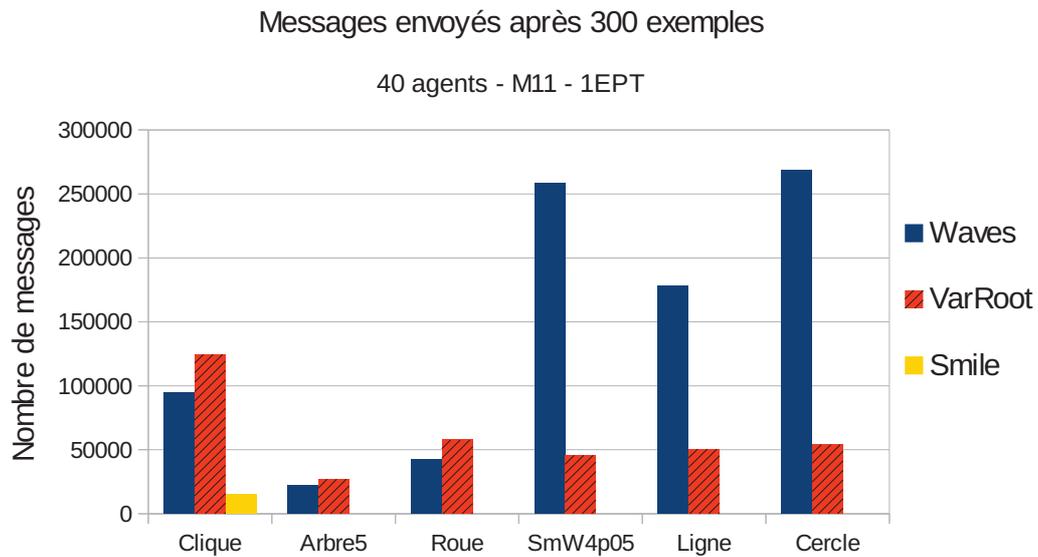
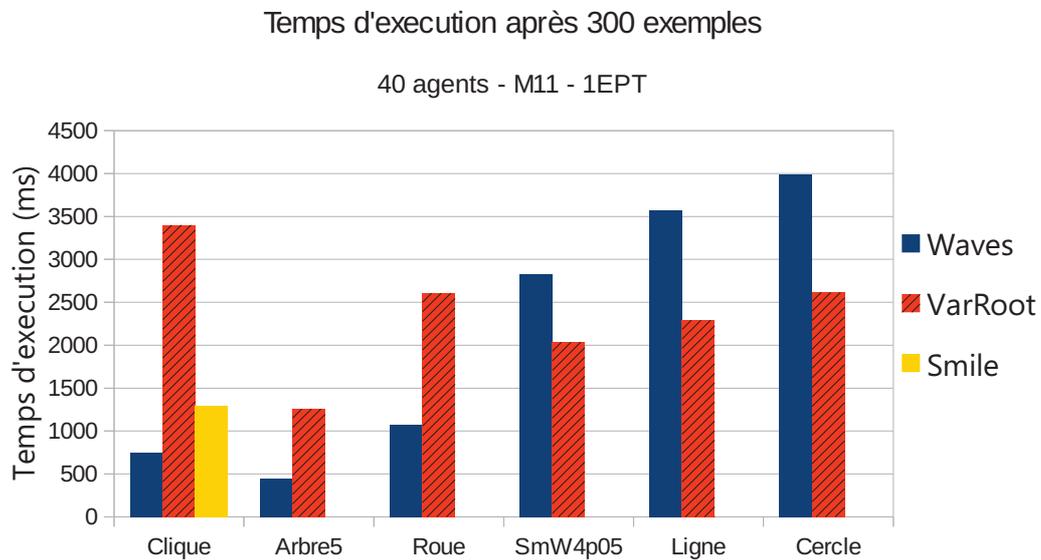


FIGURE 6.9 – Efficacité de Waves comparativement à Variable-Root et SMILE en termes de temps et de messages nécessaires pour terminer un apprentissage. Cas du problème M11 avec 40 agents limité à 1EPT évalué après 300 exemples d'apprentissages.

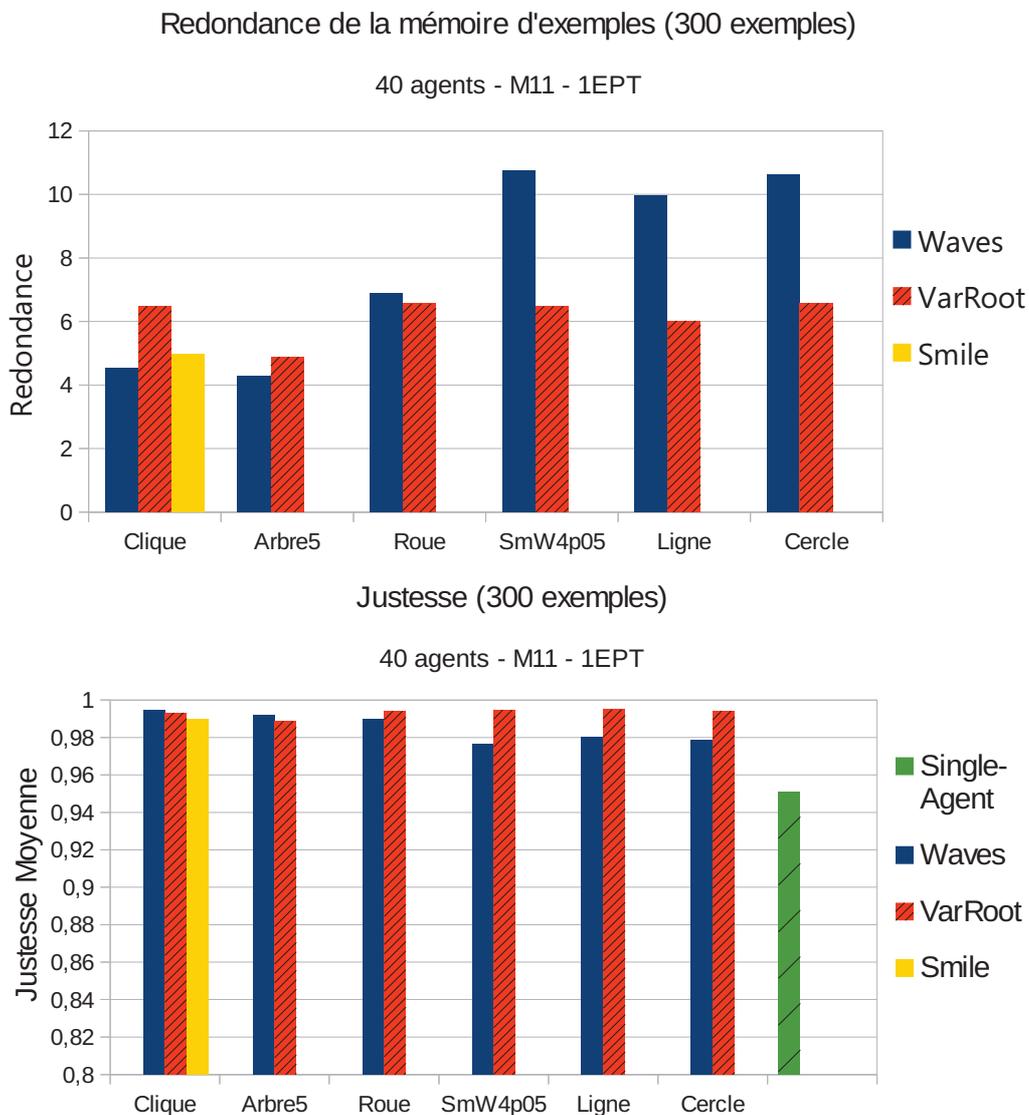


FIGURE 6.10 – Efficacité de Waves comparativement à Variable-Root et SMILE en termes de redondance de la mémoire d'exemples et de justesse. Cas du problème M11 avec 40 agents limité à 1EPT évalué après 300 exemples d'apprentissages.

avantage à Waves puisqu'il est en mesure de communiquer par broadcast et permet à plusieurs agents d'être apprentis en même temps. Ceci est cependant couteux en matière de redondance de la mémoire d'exemples et a un impact sur le gain exploratoire³ de justesse. Il peut également arriver qu'une hypothèse révisée en un point du graphe soit encore propagée ailleurs inutilement, l'information de la révision prenant du temps à être communiquée.

Les graphes centralisés avec peu de cycles, comme des arbres de hauts degrés, sont les meilleures structures pour tirer parti des forces de Waves.

Plusieurs améliorations seront présentées en partie 6.3, qui diminuent nettement les faiblesses identifiées ici, en réduisant les diverses redondances couteuses de Waves.

Influence du rythme d'acquisition des exemples et du choix d'hypothèse personnelle

Nous avons vu qu'il est possible d'avoir plusieurs hypothèses différentes qui se propagent en même temps avec le protocole Waves. Avec 1EPT le cas reste plutôt rare sauf pour des graphes présentant de longues distances et d'autant plus s'il présente également des cycles. Waves est conçu pour permettre au SMA d'acquérir un plus grand nombre d'exemples par tour. Avec 10 EPT, il est beaucoup plus fréquent d'avoir de multiples hypothèses à l'étude en même temps. De manière générale pour un même nombre d'exemples, il y a moins de tours, mais plus d'hypothèses par tour et cela demande plus de travail aux agents de les amener toutes à SMA-consistance et augmente également les redondances précédemment identifiées.

Face à un grand nombre d'hypothèses, le choix de l'hypothèse personnelle prend tout son intérêt. Parmi les trois choix proposés, le choix Shorter est toujours nettement meilleur dès qu'il y a effectivement plusieurs hypothèses à départager. On trouve peu de différence entre LastSeen et Random. La figure 6.11 montre ces effets généraux à la fois de l'augmentation d'exemples par tour, mais aussi le bénéfice du choix Shorter par rapport à LastSeen. On remarque qu'avec 1EPT le choix n'a aucun effet dans les structures Clique, Arbre5 et Roue ce qui témoigne de la rareté de la coexistence de plusieurs hypothèses dans ce cas précis. La perte de justesse à choix équivalent entre 1EPT et 10EPT s'explique par la baisse de variabilité dans l'exploration due à la plus grande redondance des exemples en mémoire. Si la Ligne bénéficie, en matière de temps d'exécution, d'un rythme plus élevé d'acquisition

3. rappel : le gain exploratoire correspond à la différence de justesse en faveur d'un SMA plutôt que d'un agent seul dans le cas de l'apprentissage d'un problème difficile.

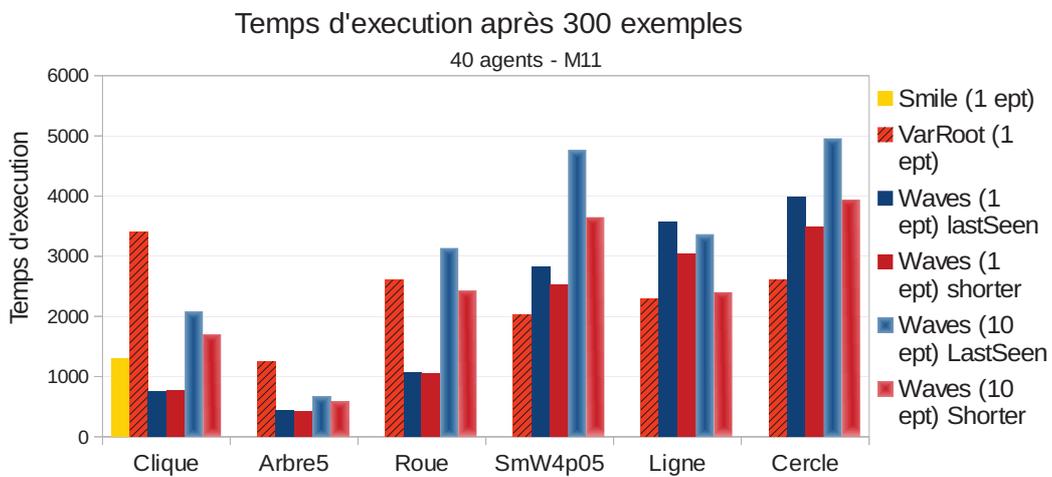
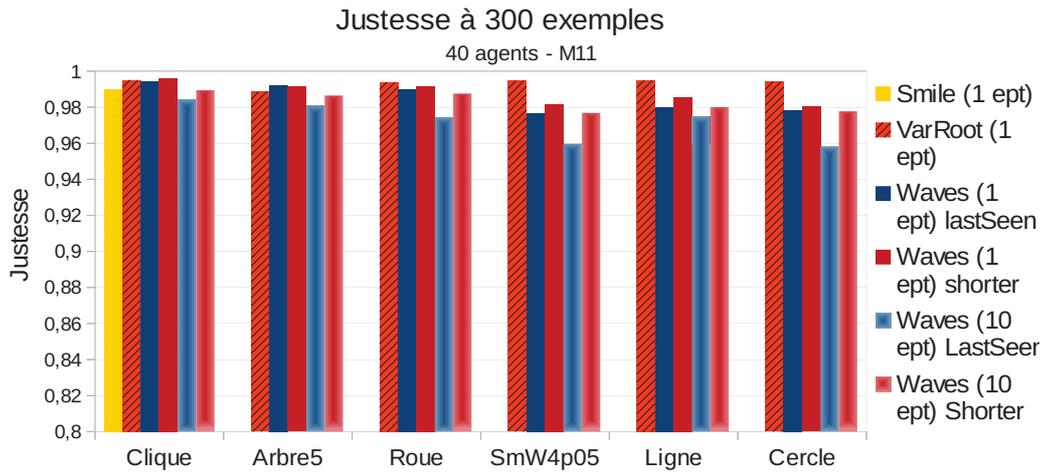


FIGURE 6.11 – Evaluation de l'efficacité de l'apprentissage selon différents choix (LastSeen-Shorter) et nombre d'exemples par tour (1-10ept)

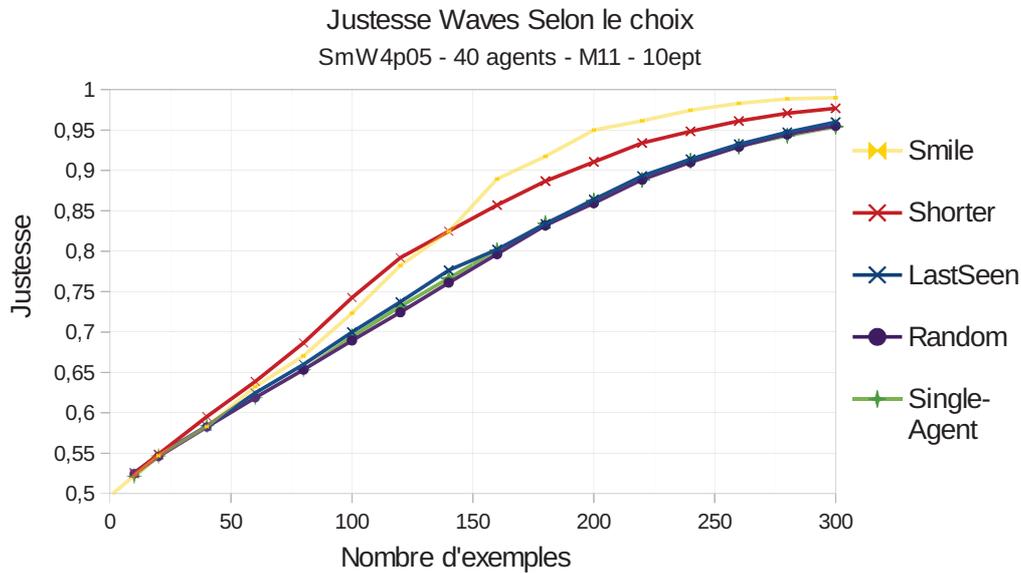


FIGURE 6.12 – Influence du choix d’hypothèse personnelle au cours de l’apprentissage.

d’exemples, c’est qu’en n’ayant aucune branche elle limite fortement l’augmentation du nombre d’hypothèses. Une hypothèse se propageant dans une Ligne aura plus souvent une fille unique révisée plusieurs fois que plusieurs filles différentes qui doivent toutes être amenées à SMA-consistance. Enfin l’augmentation de justesse du choix Shorter explique également son plus faible cout en temps, puisqu’une hypothèse de qualité a moins besoin d’être révisée et diminue donc le nombre de calculs.

Le choix personnel plus en détail

Pour détailler l’influence du choix au cours de l’apprentissage on se focalise sur le cas favorisant le plus la multiplicité des hypothèses, le Small-World avec 10EPT sur la figure 6.12. On confirme la faible différence entre LastSeen et Random tandis que Shorter prend rapidement de l’avance pendant les 10 premiers exemples et la conserve jusqu’à la fin. Afin de détailler mieux cette différence entre Shorter et LastSeen on effectue une mesure complémentaire sur l’ensemble des hypothèses vérifiées en fin de tour. On évalue la moyenne de la justesse de toutes les hypothèses de l’ensemble, représentée pour les choix LastSeen et Shorter sur la figure 6.13. Deux remarques découlent de cette mesure complémentaire :

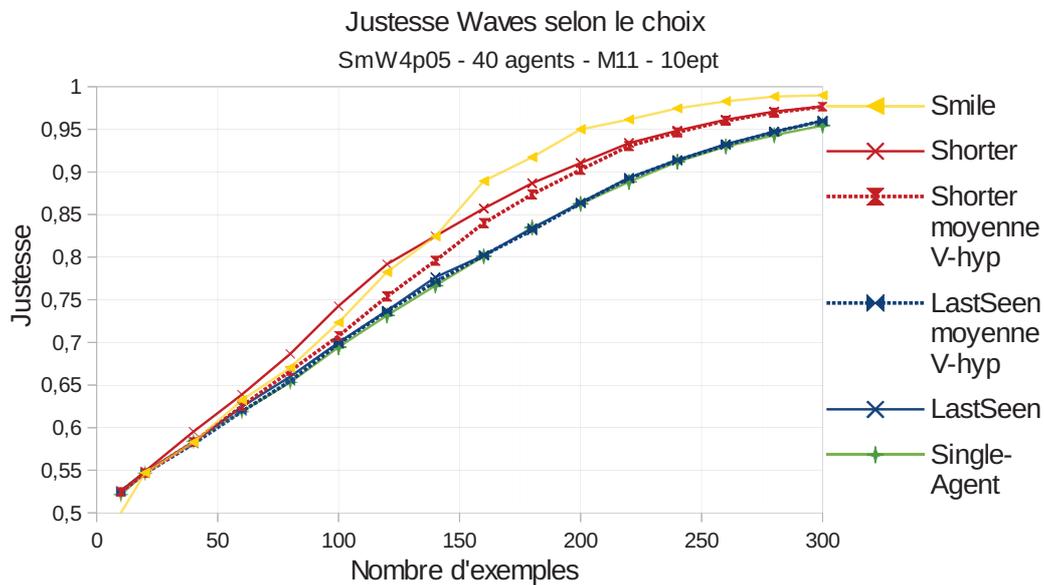


FIGURE 6.13 – Influence détaillée du choix d’hypothèse personnelle au cours de l’apprentissage. Comparaison des hypothèses choisies avec la moyenne de l’ensemble des hypothèses Vérifiées.

- **influence du choix pendant le tour** : la justesse des hypothèses choisies selon le choix Shorter est nettement meilleure que la moyenne des hypothèses générées pendant le tour. Cet écart diminue tandis que l’apprentissage progresse ce qui s’explique par le fait qu’une hypothèse proche du modèle cherché rencontre moins de contre-exemples et est donc moins souvent révisée donnant lieu à moins d’hypothèses par tour. Le choix LastSeen n’est pas particulièrement meilleur ou moins bon que l’ensemble des hypothèses vérifiées.
- **influence sur les hypothèses générées au tour suivant** : la moyenne de justesse de l’ensemble des hypothèses vérifiées avec le choix Shorter s’écarte progressivement du choix LastSeen. L’hypothèse personnelle est prioritaire en début de tour pour servir de base à une révision à l’arrivée d’un nouvel exemple en début de tour. C’est par ce moyen que le choix à une influence sur le tour suivant. Les hypothèses générées à partir d’hypothèses de meilleure qualité sont donc en moyenne également de meilleure qualité.

Bien que Waves permette toujours d’obtenir une justesse supérieure ou égale à l’agent seul, on peut souhaiter vouloir maintenir également le gain exploratoire de justesse apporté par l’apprentissage multi-agents. On a vu que

Waves tend à augmenter la redondance de la mémoire d'exemple des agents avec un impact sur l'exploration de l'espace des hypothèses et donc sur le gain exploratoire de justesse. On peut rétablir cette variabilité avec une option de SMILE particulièrement efficace en présence de broadcast, l'oubli des exemples externes présenté en 3.1.4. L'oubli dans Waves ne peut être effectué qu'en fin de tour ou plus précisément quand un agent prend connaissance du début d'un nouveau tour par l'arrivée d'une nouvelle information de date plus récente. Cet oubli permet d'obtenir une bien meilleure précision comme l'on peut voir figure 6.14. Cette option n'est pas à prendre à la légère, l'oubli dans un réseau est beaucoup plus couteux que dans une clique. Le temps d'exécution pour le cas du graphe SmW4p05 à 10EPT avec choix Shorter passe de 3,6 sec à 13,2 sec soit presque 4 fois plus. Pour le Cercle ou la Ligne ces temps augmentent encore plus considérablement.

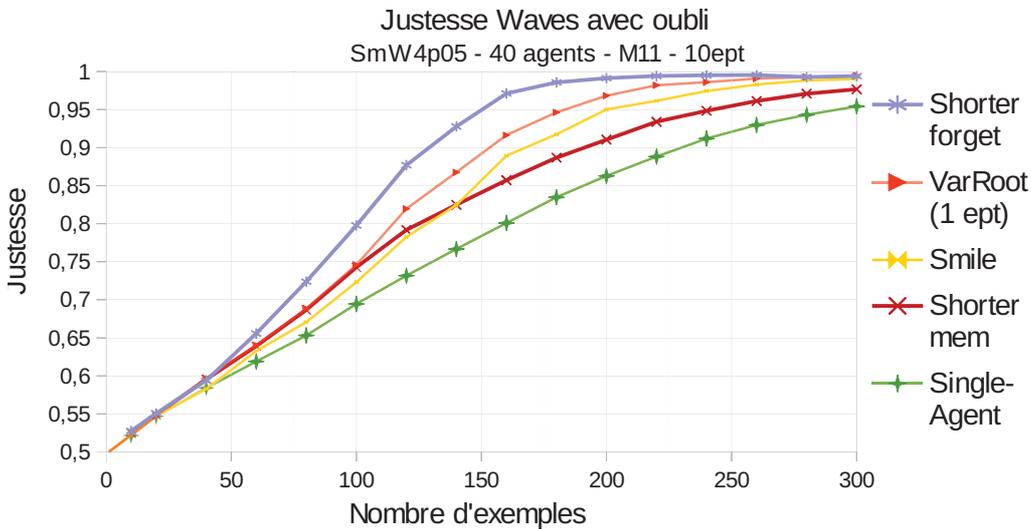


FIGURE 6.14 – L'oubli des exemples externes à une influence positive sur le gain exploratoire de justesse par rapport à l'agent seul.

Note : Il est possible d'augmenter encore le nombre d'exemples par tour. On conserve ici une mémorisation totale des exemples. Le premier tour permet d'atteindre une justesse supérieure à celle obtenue avec un même nombre d'exemples dont l'arrivée se fait à un rythme moins soutenu. La progression est ensuite ralentie et s'éloigne de la courbe obtenue avec oubli pour se rapprocher de celle de l'agent seul.

6.2.3 Bilan du protocole Waves initial

Waves fonctionnel. Le protocole Waves atteint les objectifs fixés de donner à l'agent une vision locale tout en maintenant des garanties de SMA-consistance et de terminaison. Il permet une propagation dynamique et multiple des hypothèses et de leur révisions. On peut envisager différents critères de choix d'hypothèse personnelle. La justesse des hypothèses choisies par les agents est toujours au moins égale à celle d'un agent apprenant seul tous les exemples. Il bénéficie en partie du gain exploratoire d'apprentissage multi-agents sur les problèmes difficiles et permet d'acquérir plusieurs exemples à la fois, répartis dans le graphe.

Waves redondant Le protocole Waves présente encore de nombreuses formes de redondances qui sont coûteuses en matière de calculs effectués, messages envoyés et informations mémorisées. On distingue au moins trois formes de redondances sur lesquelles on peut proposer des améliorations qui seront détaillées dans la partie 6.3 .

- Certaines hypothèses sont révisées plus que nécessaire. Le choix de la priorité LIFO sur les hypothèses à traiter permet de limiter le nombre d'agents devant traiter une hypothèse qui est déjà démontrée non SMA-consistante. Un problème similaire n'est cependant pas encore pris en compte. Certaines hypothèses identiques sont formées en différents endroits du graphes, mais possédant des identifiants différents. Elles sont donc traitées indépendamment générant des communications et **calculs redondants pour les apprentis**. Il faut envisager une gestion de classes d'équivalence d'hypothèses.
- La vision locale des agents ne leur permet pas de garder une trace des agents ayant critiqué une hypothèse. Une hypothèse peut être proposée de nombreuses fois à un même agent qui doit alors en vérifier la consistance avec sa mémoire d'exemple. On souhaite donc limiter la **redondance des critiques**.
- Le broadcast et la multiplicité des hypothèses augmente la **redondance de la mémoire d'exemple**.

Les deux premières redondances impactent la charge de calcul et donc le temps d'exécution tandis que la troisième a une influence sur la justesse obtenue.

6.3 Améliorations

On propose des mécanismes pour limiter les redondances identifiées dans la partie précédente.

6.3.1 Fusion d'hypothèses par classe d'équivalence

On propose de définir des classes d'équivalence d'hypothèses qui regroupent toutes les hypothèses composées des mêmes termes. En considérant les classes d'équivalence, il existe désormais une ambiguïté entre hypothèse au sens d'identifiant unique et hypothèse au sens de contenu logique avec classe d'équivalence d'identifiants. Le théorème 1 est à considérer au sens d'une hypothèse par identifiant. On nommera une hypothèse avec classe d'équivalence eq-hypothèse.

La mise en place de classes d'équivalence pour les hypothèses s'est effectuée en deux temps.

Une première tentative

On groupe d'une part les identifiants et d'autre part les ancêtres.

Exemple 11 fusion selon première version :

$H_1 : (date_1, \{id_1\}, \{ancêtres_1\}, cont_logique_1)$

$H_2 : (date_1, \{id_2\}, \{ancêtres_2\}, cont_logique_1)$

$H_{fusion} : (date_1, \{id_1, id_2\}, \{ancêtres_1\} \cup \{ancêtres_2\}, cont_logique_1)$

Problème 1 : Il peut alors arriver qu'une eq-hypothèse soit invalidées par une descendante d'une partie de sa classe d'équivalence seulement et la propagation des ancêtres de l'autre partie n'est plus assurée. La propriété $p6^4$ n'est plus vérifiée.

Solution 1 : On peut alors rajouter un message qui informe des identifiants rejetés qui ne peuvent plus être portés par une eq-hypothèse vérifiée.

Problème 2 bloquant : Cette solution rétabli la propriété $p6$, mais invalide la propriété $p4^5$. Cette information peut invalider des eq-hypothèses

4. $p6$: Un agent doit être informé si une hypothèse qu'il a traitée a été contredite et la retirer alors de ses "hypothèses valides".

5. $p4$: Une hypothèse, formée pendant le tour, est traitée par tout agent de proche en proche dans le graphe jusqu'à ce qu'elle soit contredite par un contre-exemple où que sa consistance soit vérifiée par chaque agent relativement à ses exemples

avant qu'elles ne forment de fille. Or il est possible qu'une hypothèse soit logiquement équivalente à une de ses descendantes. Il est alors possible qu'un tour se finisse sans qu'il n'y ait plus aucune hypothèse vérifiée.

Résultats : Cette version ne permet pas de maintenir les garanties de Waves et est donc abandonnée. En pratique elle est cependant prometteuse sur l'intérêt des classes d'équivalence. La perte de toute hypothèse reste un cas rare. Il est possible d'obtenir un gros gain de temps d'exécution sans modification de la justesse obtenue. Par exemple SmW4p05 à 1EPT passe de 2500ms à 1200ms. Les structures et conditions (ex Arbre5 à 1EPT) identifiées comme ne donnant lieu à la formation que d'une seule hypothèse par tour ne sont bien évidemment pas affectées par ce changement.

La version définitive

Principe. On abandonne l'idée du message supplémentaire de la solution 1. La nouvelle solution qui évite le problème 1 maintient cette fois-ci le lien entre un identifiant et ses ancêtres pour grouper dans une classe d'équivalence ces couples d'informations qu'on appellera *passport* :

Définition 11. Un *passport* est un couple (id, Anc) où :

- id est l'identifiant de l'hypothèse associée au *passport*
- Anc est l'historique de l'hypothèse du *passport*, c'est-à-dire l'ensemble des identifiants de ses ancêtres

Pour un *passport* p , on notera $p.id$ son identifiant et $p.Anc$ son historique.

Un *passport* est invariable après sa création. Avec cette solution, une hypothèse descendante qui n'invalidé qu'une partie des identifiants permet d'identifier quelle partie de la classe d'équivalence n'a pas été suffisamment propagée et donc de poursuivre sa propagation avec les *passports* qui n'ont pas été invalidés. L'hypothèse est propagée avec ses *passports* restant jusqu'à ce qu'elle soit révisée de manière à garantir les propriétés $p6$ et $p4$.

Définition 12. Une eq-hypothèse Waves est constituée de :

- une date
- un ensemble de *passports*
- un contenu logique

Ils seront respectivement notés $H.date$, $H.passeports$ et $H.cont_logique$ pour une hypothèse H .

Exemple 12 fusion selon la version définitive :

$H_1 : (date_1, \{(id_1, \{ancêtres_1\})\}, cont_logique_1)$

$H_2 : (date_1, \{(id_2, \{ancêtres_2\})\}, cont_logique_1)$

$H_{fusion} : (date_1, \{(id_1, \{ancêtres_1\}), (id_2, \{ancêtres_2\})\}, cont_logique_1)$

Algorithme. Le pré-traitement (réception d'une hypothèse par message de groupe-consistance (H_{new})) est notablement complexifié par la gestion des classes d'équivalences. Le filtre de date n'est pas modifié. Le filtre de l'identifiant par contre doit déterminer non seulement si l'hypothèse est à traiter (algorithme 1), mais aussi de quelle manière (algorithme 2). Enfin, il faut adapter aux passeports multiples le filtre des ancêtres (algorithme 3). On ne peut en effet pas se contenter d'effectuer un simple retrait d'une hypothèse ayant un identifiant compris dans les ancêtres de l'hypothèse nouvellement reçue. L'hypothèse courante est notée H_{curr} et \equiv représente l'équivalence terme à terme du contenu logique de deux eq-hypothèses.

Algorithm 1 Filtre d'identifiant (1/2) : rejet des hypothèses déjà traitées

```

 $H_{new.passeports} \leftarrow \{p \in H_{new.passeports} / p.id \notin H\_Rejetées\}$ 
if  $H_{new.passeports} = \emptyset$  then
  Fin prétraitement
end if
for all  $H \in H\_Vérifiées \cup H\_ATraiter \cup \{H_{curr}\}$  do
  if  $(H_{new.passeports} \subseteq H.passeports)$  then
    Fin prétraitement
  end if
end for

```

Lorsqu'un agent forme une nouvelle hypothèse par révision, il doit également vérifier s'il peut la placer dans la classe d'équivalence d'une de ses eq-hypothèses en mémoire. Il procède de manière similaire à l'algorithme 2.

Résultats

L'utilisation des classes d'équivalence permet de diminuer le nombre d'hypothèses à proposer et réviser. On constate que cette amélioration **compense la présence de cycles**. En effet dans un cycle une hypothèse peut souvent arriver au contact du même agent détenant le contre exemple par les chemins

Algorithm 2 Filtre d'identifiant (2/2) : fusionner si possible et déterminer comment traiter l'hypothèse

```

hypFound ← false
if apprentice then
  if  $H_{curr} \equiv H_{new}$  then
     $H_{curr}.passeports \leftarrow H_{curr}.passeports \cup H_{new}.passeports$ 
    AAbandonner ← false
    hypFound ← true
  end if
end if
if not(hypFound) then
  for all  $H \in H\_ATraiter$  do
    if  $H \equiv H_{new}$  then
       $H.passeports \leftarrow H.passeports \cup H_{new}.passeports$ 
      hypFound ← true
      break
    end if
  end for
end if
if not(hypFound) then
  for all  $H \in H\_Vérifiées$  do
    if  $H \equiv H_{new}$  then
       $H.passeports \leftarrow H.passeports \cup H_{new}.passeports$ 
      broadcast Group – Consistance( $H$ )
      hypFound ← true
      break
    end if
  end for
end if
if not(hypFound) then
  if apprentice then
     $H\_ATraiter \leftarrow H\_ATraiter \cup \{H_{new}\}$ 
  else
    Traitement  $H_{new}$ 
  end if
end if

```

Algorithm 3 Filtre sur les ancêtres de H_{new}

```
 $Anc_{new} \leftarrow (\bigcup_{p \in H_{new}.passeports} p.Anc) \setminus H\_Rejetées$   
 $H\_Rejetées \leftarrow H\_Rejetées \cup Anc_{new}$   
#Mettre à jour les autres ensembles d'hypothèses :  
 $H_{curr}.passeports \leftarrow \{p \in H_{curr}.passeports / p.id \notin Anc_{new}\}$   
if  $H_{curr}.passeports = \emptyset$  then  
   $AAbandonner = true$   
end if  
for all  $H \in H\_ATraiter$  do  
   $H.passeports \leftarrow \{p \in H.passeports / p.id \notin Anc_{new}\}$   
  if  $H.passeports = \emptyset$  then  
     $H\_ATraiter \leftarrow H\_ATraiter \setminus \{H\}$   
  end if  
end for  
for all  $H \in H\_Vérfiées$  do  
   $H.passeports \leftarrow \{p \in H.passeports / p.id \notin Anc_{new}\}$   
  if  $H.passeports = \emptyset$  then  
     $H\_Vérfiées \leftarrow H\_Vérfiées \setminus \{H\}$   
  end if  
end for
```

de part et d'autre du cycle. Cette hypothèse est alors révisée à partir du même contre exemple selon deux agents différents avec un risque élevé de former deux hypothèses équivalente. Il est donc probable qu'elles puissent être ensuite fusionnées. La figure 6.15 montre cet effet sur les différentes structures. On remarque que Small-World, Ligne et Cercle ont ainsi des temps similaires à 1EPT. Les multiples hypothèses créées par tour dans le cas 1EPT sont souvent très similaires et dues aux cycles. La fusion permet donc au Cercle d'égaliser la Ligne.

Dans tous les cas, la fusion a un effet négligeable sur la justesse obtenue. Si l'on mesure les couts pour les apprentis et pour les critiques en matière de nombre de subsomptions effectuées, on constate sur la figure 6.16 que l'activité des apprentis est peu impactée tandis que celle des critiques reflète les variations de temps d'exécution. En effet à part un léger effet sur de très longues distances, comme dans le cas de la Ligne, les agents dans leur rôle d'apprenti effectuent presque autant de révisions avec que sans fusion alors que, en regroupant les hypothèses, ils les proposent moins souvent. Les agents dans leur rôle de critique ont donc très nettement moins de calculs à effectuer. Cette variation est d'autant plus remarquable que les couts sont près de 100

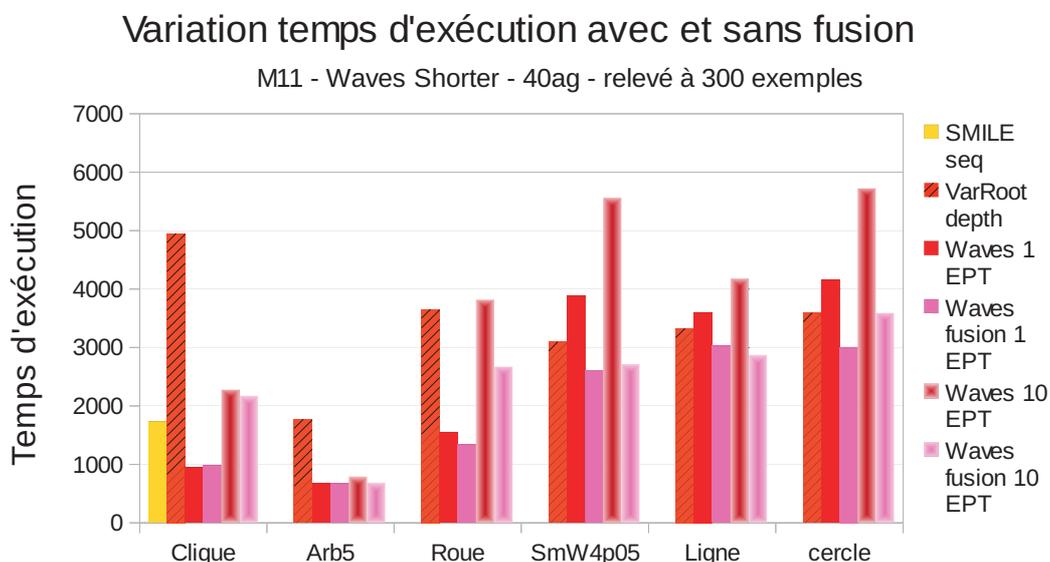


FIGURE 6.15 – Influence de la fusion d’hypothèses par classe d’équivalence sur le temps d’exécution selon les différentes topologies de réseau.

fois plus importants pour l’activité de critique que pour celle d’apprenti. On observe une diminution de plus de 50% des subsomptions pour le Cercle et le Small-World ; ils sont suivis de près par la Ligne également.

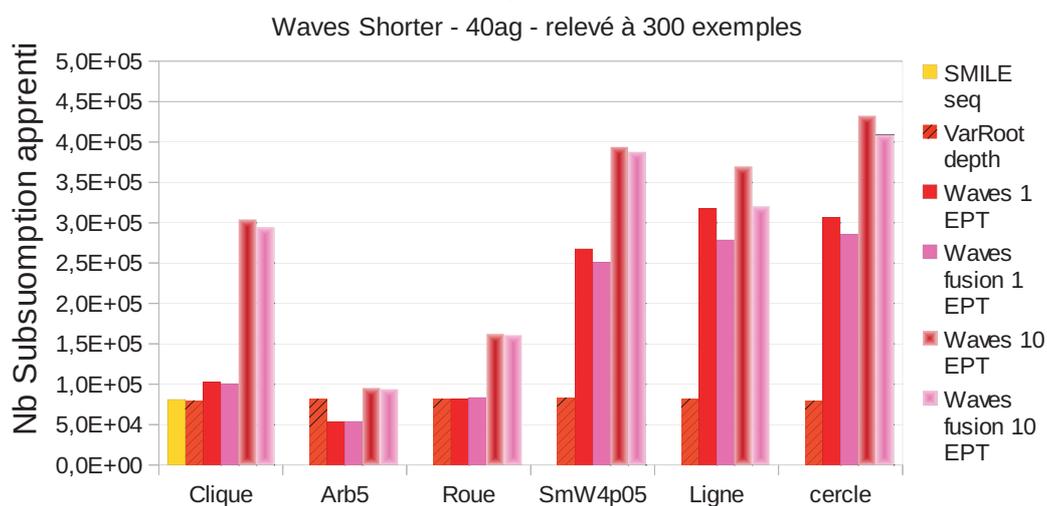
La fusion ne peut avoir un effet qu’en présence de multiples hypothèses identiques au sein d’un même tour. Les hypothèses identiques sont principalement générées par la présence de cycles et dans une moindre mesure la présence de longues distances. La structure d’arbre ne présente aucune de ces caractéristiques et n’est donc pas impactée par la fusion d’hypothèses équivalentes.

6.3.2 Économie de critiques

Principe

La fusion d’hypothèses équivalentes permet d’éviter qu’un même apprenti propose plusieurs fois une même hypothèse aux mêmes critiques. Il est possible de diminuer également le nombre de fois qu’une même hypothèse est proposée à un même agent critique par différents agents apprentis. Pour cela on rajoute une information à l’hypothèse, la trace des critiques qui sont les agents qui ont accepté cette hypothèse.

Variation coûts calculs apprenenti avec et sans fusion



Variation coûts calculs critiques avec et sans fusion

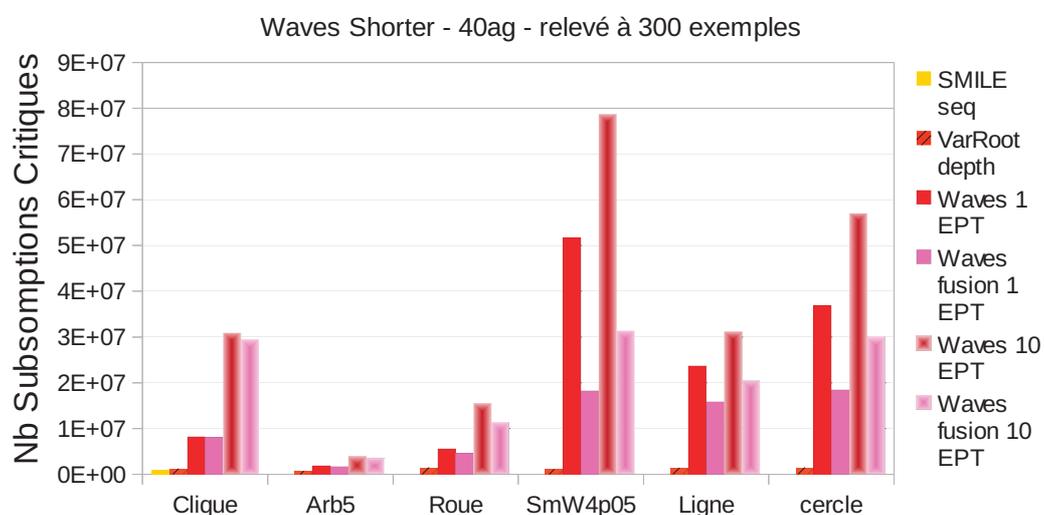


FIGURE 6.16 – Influence de la fusion d’hypothèses par classe d’équivalence sur le nombre de subsumptions effectuées par les agent selon leur rôle et selon les différentes topologies de réseau.

En considérant à la fois la fusion par classe d'équivalence et la trace des critiques effectuées, on peut se contenter de proposer une hypothèse uniquement aux voisins qui n'ont critiqué aucun des identifiants de l'hypothèse, par contre il faut toujours notifier la groupe-consistance à l'ensemble des voisins pour une bonne gestion des ancêtres. Un voisin peut avoir reçu un contre-exemple depuis qu'il a critiqué une hypothèse et donc être ignoré par un apprenti alors qu'il dispose d'un contre-exemple. Ce cas n'est cependant pas gênant puisque l'hypothèse sera confrontée à tous les agents jusqu'à rencontrer un contre-exemple. Même s'il s'agit du seul contre-exemple pour cette hypothèse, elle y sera confronté en atteignant l'agent qui a reçu cet exemple en premier. On peut donc se contenter de fusionner les traces de critiques lors d'une fusion d'hypothèses.

Exemple 13 fusion selon version définitive :

$$H_1 : (date_1, \{[id_1, \{ancêtres_1\}]\}, \{critics_1\}, cont_logique_1)$$

$$H_2 : (date_1, \{[id_2, \{ancêtres_2\}]\}, \{critics_2\}, cont_logique_1)$$

$$H_{fusion} : (date_1, \{[id_1, \{ancêtres_1\}], [id_2, \{ancêtres_2\}]\}, \{critics_1\} \cup \{critics_2\}, cont_logique_1)$$

Cette amélioration relativement peu coûteuse en matière d'opérations supplémentaires à effectuer permet de diminuer considérablement le nombre de messages échangés et le nombre de critiques. En particulier cette économie touche toutes les structures en mémorisant parmi les autres critiques le nœud père par lequel l'agent a reçu cette hypothèse. Les cycle minimaux de 3 agents sont également couverts par cette économie de critiques comme on peut le voir dans l'exemple 14.

Les cycles plus longs ne bénéficient pas de cette amélioration de manière systématique. Variable-Root en fixant un arbre a priori, maximise l'économie de critique, mais ne permet pas de s'adapter dynamiquement à la disponibilité des agents pour transmettre une hypothèse de manière optimale. On peut envisager des méthodes complémentaires pour réduire encore le nombre de critiques de Waves, mais le compromis pourrait s'avérer coûteux en matière de messages envoyés où en opérations supplémentaires.

Exemple 14 Économie de messages dans un cycle élémentaire :

Un agent A_T a deux voisins, A_{S1} et A_{S2} qui sont également reliés entre eux. Lorsque A_T notifie la **groupe-consistance**(H) il inclut dans l'hypothèse H l'information des agents qui l'ont déjà acceptée. A_{S1} et A_{S2} sont inclus dans la liste des critiques ainsi que A_T lui-même. Les voisins n'auront donc pas à envoyer **propose**(H) ni l'un à l'autre ni à A_T comme sur la figure 6.17. On économise ainsi les messages **propose** et **accept**, mais surtout aussi le cout

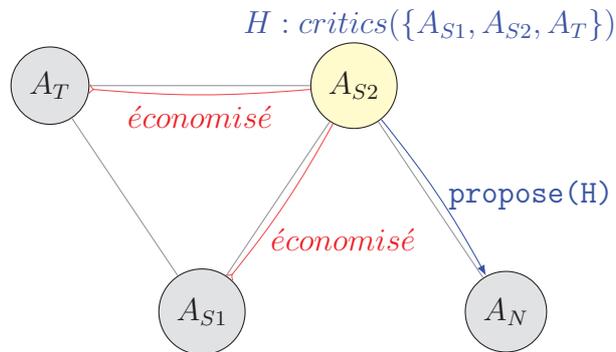


FIGURE 6.17 – Avec la trace des critiques déjà consultés, un apprenti peut économiser des messages `propose` et ses voisins peuvent éviter de tenir de nouveau un rôle de critique qu'ils ont déjà rempli.

de la vérification de H avec les exemples en mémoire quatre fois dont les deux nécessairement successives de A_T .

Résultats

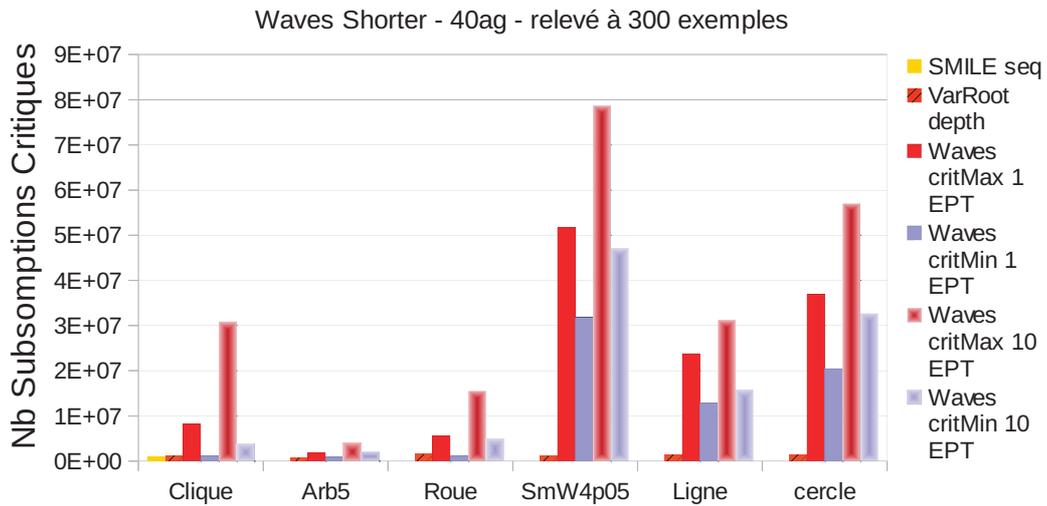
Les Résultats de la figure 6.18 confirment l'importance de l'effet bénéfique de l'économie de critique (CritMin) par rapport à la version de Waves initiale (CritMax). La figure montre un effet important pour tous les graphes et encore plus marqué pour la Clique et la Roue qui comportent de très nombreux cycles élémentaires. Cette amélioration a un effet négligeable sur la justesse et les subsumptions des agents dans leur rôle d'apprenti, mais est très marqué concernant les subsumptions du rôle de critique.

Diminution de critiques complémentaires

Les deux améliorations proposées de la fusion d'hypothèses par classe d'équivalence et de la trace des critiques conservée dans l'hypothèse sont compatibles et complémentaires. L'utilisation de ces deux options permet d'améliorer encore plus les temps d'exécution et nombre de critiques pour bénéficier des points forts de Waves en ayant fortement réduit ses faiblesses. Comme on peut le voir figure 6.19, il bénéficie ainsi suffisamment du traitement en parallèle pour être plus efficace en matière de temps d'exécution que Variable-Root⁶ dans toutes les situations. On remarque également dans

6. Waves est plus efficace que les deux versions de Variable-Root bien que celle utilisant une construction en profondeur soit la seule représentée.

Variation coûts calculs critiques avec limitation de critiques



Variation temps d'exécution avec limitation de critiques

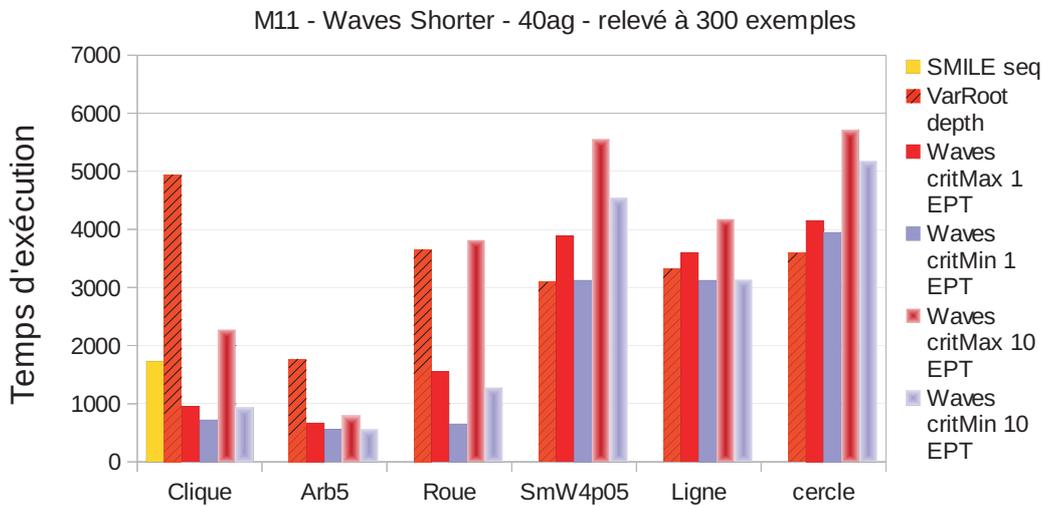


FIGURE 6.18 – Influence de la trace des critiques d’une hypothèse, sur le nombre de subsumptions effectuées par les agents dans un rôle de critiques et sur le temps d’exécution, selon les différentes topologies de réseau.

Temps d'exécution avec fusion et limitation de critiques

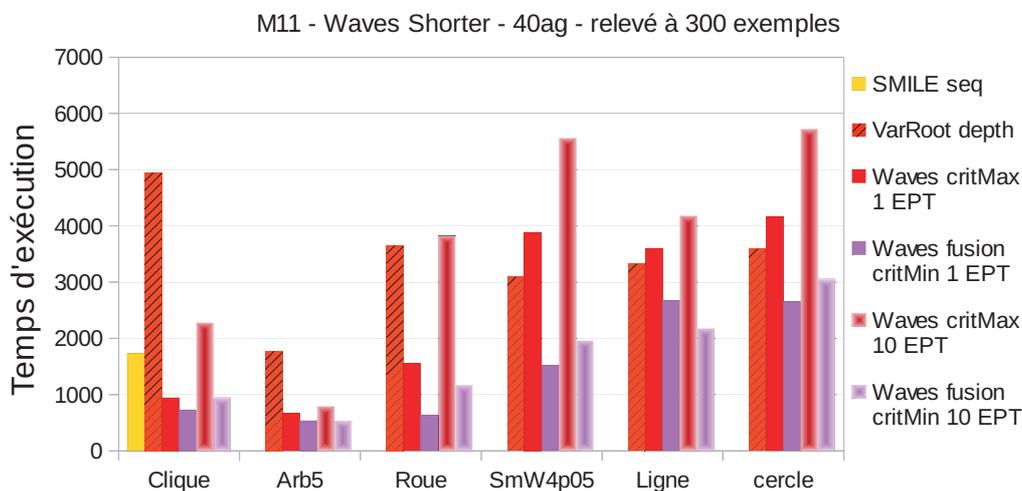


FIGURE 6.19 – Influence de l'utilisation conjointe de la trace des critiques et de la fusion d'hypothèses, sur le temps d'exécution, selon les différentes topologies de réseau.

le cas de la Ligne que l'arrivée simultanée de 10 exemples par tour est traité plus rapidement qu'une arrivée un à un des exemples.

6.3.3 Oubli

Parmi les redondances identifiées, seule celle concernant la mémoire d'exemples a un impact sur la justesse. Le moyen envisagé pour diminuer cette redondance et rétablir le gain exploratoire de justesse atteint par SMILE, est d'utiliser un oubli des exemples externes en fin de révision globale (fin de tour pour Waves). On conserve dans tous les cas les exemples internes afin d'assurer la convergence du protocole. Un oubli dit total concerne donc seulement les exemples externes.

L'oubli en présence d'agents tous connectés

Comme vu en section 3.1.4, le protocole SMILE avec sa version broadcast et oubli est particulièrement efficace. Nous avons testé des formats de mémoire d'exemples différents en clique au moyen de bornes et de filtres.

On peut voir une borne sur la mémoire comme une condition d'oubli d'une certaine partie des exemples. On recherche, en l'imposant aux agents, une

diminution des couts d'apprentissage (temps, messages, calculs) sans perte de qualité. En particulier moins d'exemples en mémoire signifient, pour un agent critique, moins de tests à effectuer sur l'hypothèse proposée et, pour un apprenti, moins d'exemples à tenir compte lors d'une révision.

Plusieurs versions de mémoires bornées : on définit la mémoire des agents selon trois blocs, la mémoire centrale, la mémoire temporaire et la mémoire supplémentaire. Toutes les versions considérées ont les mêmes mémoires centrales et temporaires mais se distinguent par un fonctionnement différent de leur mémoire supplémentaire. Les agents mémorisent sans aucune limite, dans la mémoire centrale (core memory), tous les exemples internes qui ont été des contre-exemples pour cet agent. Ils mémorisent tous les exemples externes dans la mémoire temporaire (temporary memory), sans limite de place, mais uniquement le temps d'une révision. La mémoire supplémentaire (sample memory) peut accueillir, selon ses restrictions, les exemples internes qui n'ont jamais servi de contre-exemple et les exemples externes qui sont retirés de la mémoire temporaire. Elle est limitée en taille et fonctionne de manière circulaire de manière à privilégier les exemples les plus récents. En fin de révision pour SMILE et fin de tour pour Waves, il est possible de déclencher un oubli des exemples externes. Avec cette définition de la mémoire, l'oubli devient une remise à zero de la mémoire temporaire. Les exemples externes qu'elle contenait peuvent alors être sauvegardés dans la mémoire supplémentaire si elle dispose d'assez de place et que la priorité leur est favorable. Les priorités et restrictions envisagées sur la mémoire supplémentaire sont les suivantes selon le protocole :

- Layered : La priorité est donnée aux exemples les plus récemment acquis par l'agent indifféremment de leur origine (internes ou externes).
- Lay-Prio-Int : la priorité est donnée aux exemples internes. (Un exemple externe peut être refusé ou retiré pour faire de la place à un exemple interne. Un exemple interne peut être rejeté uniquement si la mémoire est pleine d'exemples internes.)
- Lay-Prio-CE : la priorité est donnée aux contre-exemples.
- Lay-Filt-Int : filtre les exemples de manière à ne mémoriser que les internes. (Aucun exemple externe n'est conservé)

Propriétés et objectifs. La mémoire centrale garantit la convergence de l'apprentissage de chaque agent. La mémoire temporaire garantit la terminaison de chaque révision et donc du protocole.

En faisant varier la troisième mémoire, le but est d'identifier l'importance de conserver les exemples internes, qui ne sont pas possédés par aucun autre agent mais n'ont pas eu de rôle actif sur l'apprentissage, et les exemples externes, qui ont provoqué des révisions par le passé. On teste ainsi un oubli sélectif avec différentes versions de mémoire bornées sur les deux protocoles séquentiel et broadcast de SMILE. Par comparaison, les protocoles de références ont une mémoire simple, non bornée et effacent uniquement les exemples externes lors d'un éventuel oubli.

Résultats en séquentiel :

- En faisant varier la taille maximale de la mémoire, on peut trouver un pic de performances (temps plus court, moins de sousomptions, moins de messages, mais justesse équivalente) par rapport aux deux protocoles de référence SMILE séquentiel avec et sans oubli des exemples externes. Certaines de ces caractéristiques (temps, sousomption, nb message, justesse) ne sont pas toujours meilleures en fonction de la taille de la mémoire supplémentaire.
- Le pic de performances correspond à une taille maximum de la mémoire supplémentaire d'autant plus petite qu'il y a plus d'agents. Ce qui semble naturel vu qu'une même taille de mémoire individuelle implique plus d'exemples mémorisés dans le système lorsque les agents sont plus nombreux.
- Les exemples internes sont importants :
 Une mémoire supplémentaire nulle (taille 0) a de moins bon résultats que les deux protocoles de référence. La différence de stockage avec SMILE oubli concerne les exemples internes non CE.
 Lay-Filt-Int est équivalent à SMILE oubli dès que la taille de la mémoire interne est suffisante pour stocker tous les exemples internes, il est moins performant dès que la borne de la mémoire lui fait perdre des exemples Internes.
- L'écart entre Lay-Filt-Int et Lay-Prio-Int montre que la présence d'un peu de contre-exemples externes peut être bénéfique en début d'apprentissage.
- Lay-Prio-Int > Layered > Lay-Prio-CE semble être presque toujours vérifié, Il semble plus intéressant de donner la priorité à la diversité plutôt qu'aux contres exemples.

Résultats en broadcast Le protocole SMILE broadcast avec oubli total des exemples externes est supérieur en tous points à la version séquentielle, mais aussi à toutes les versions de mémoires bornées basées sur SMILE broadcast. Ses couts sont plus faibles en nombre de subsomptions pour l'apprenti et les critiques ainsi qu'en temps et nombre de messages.

L'oubli en réseau

Dans un réseau l'oubli est d'autant plus couteux que la transmission d'un exemple d'un agent à son voisin a lieu uniquement lors d'une révision d'hypothèse et lorsque les distances entre agents ne sont plus élémentaires. Pour qu'un exemple soit transmis sur de longues distances, beaucoup de calculs sont donc nécessaires. La variabilité de l'exploration a rapidement un cout très élevé lorsque la distance moyenne entre paires d'agents augmente. On cherche donc un compromis acceptable entre la justesse et le temps d'exécution au moyen d'oubli plus raisonné que tout ou rien.

On considère, entre un oubli total des exemples externes et aucun oubli, de n'oublier que les exemples externes proches (distance 1, distance 1 à 2, distance 1 à 3) ou encore que les exemples provenant d'une distance impaire (dist. impaire). Le critère de distance impaire permet d'oublier un grand nombre d'exemples tout en les conservant à distance 1 dans la mémoire d'au moins un des voisins. Les tests sont effectués sur Waves avec les meilleures options obtenues jusqu'à présent (LIFO - Shorter - Fusion - CritMin).

La remarque la plus évidente est que le cout des diverses options intermédiaires d'oubli reste raisonnable, contrairement à l'oubli total (figure 6.20). Le cout en matière de temps d'exécution ne dépasse pas le double du temps de Waves sans oubli sauf pour l'oubli des exemples à faible distance dans une Roue.

Les différences entre 1EPT et 10EPT concernent principalement les formes d'oubli extrêmes et les topologies de grand diamètre. Sans aucun oubli le protocole permet d'atteindre une meilleure justesse à 1EPT pour la Ligne et le Cercle. Sur ces deux même graphes le protocole avec un oubli total nécessite près de deux fois moins de temps et il atteint une justesse proche de 0,995 pour toutes les topologies à 10EPT. Le protocole avec oubli total est le seul à bénéficier d'un plus grand nombre d'exemples par tour. La plupart des modes d'oubli sont plus performant à 1EPT. À part pour le Small-World, un oubli à distance faible semble être inutile à 1EPT. Un oubli à distance faible est cependant moins affecté par le passage à 10EPT que la version sans oubli et, pour les longs diamètres, il présente une meilleure justesse pour un temps légèrement plus court.

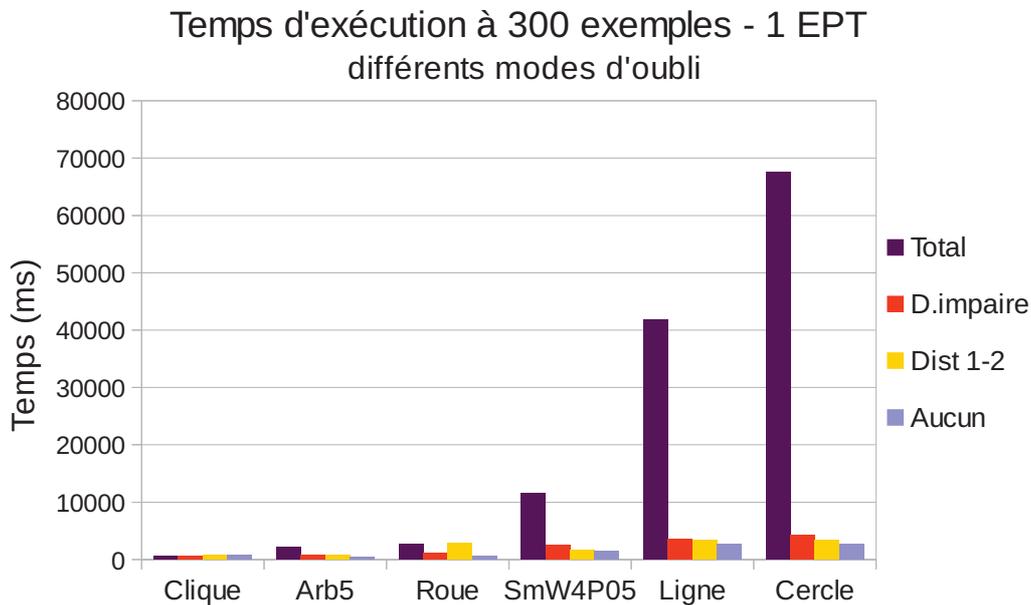


FIGURE 6.20 – Comparaison des temps d'exécution pour l'apprentissage du M11, après 300 exemples dans différentes topologies de 40 agents, avec le protocole Waves et différentes formes d'oubli des exemples externes. Le temps d'exécution est plus raisonnable pour les formes d'oubli non totales.

On remarque, sur les figures 6.21 et 6.22, que l'oubli à distance impaire est beaucoup plus indépendant de la structure du réseau qu'un oubli à distance fixé. Pour un graphe de long diamètre, un oubli des exemples à distance faible, donne des résultats très proches de aucun oubli, tandis que, pour des graphes de très faibles diamètre, il est similaire à un oubli total.

Le compromis, le plus intéressant obtenu par cette étude, est celui de l'oubli à distance impaire. Il permet d'atteindre une justesse de 0,99 avec toutes les topologies à 300 exemples pour un temps d'exécution entre 1 et 1,7 fois celui de Waves sans oubli.

Les résultats de l'oubli total des exemples externes à 10EPT peuvent encourager à chercher d'autres compromis qui amélioreraient la justesse, mais le manque à gagner n'est plus très conséquent. Une piste est d'envisager un oubli moins fréquent, mais un peu plus important afin de maintenir une diversité exploratoire.

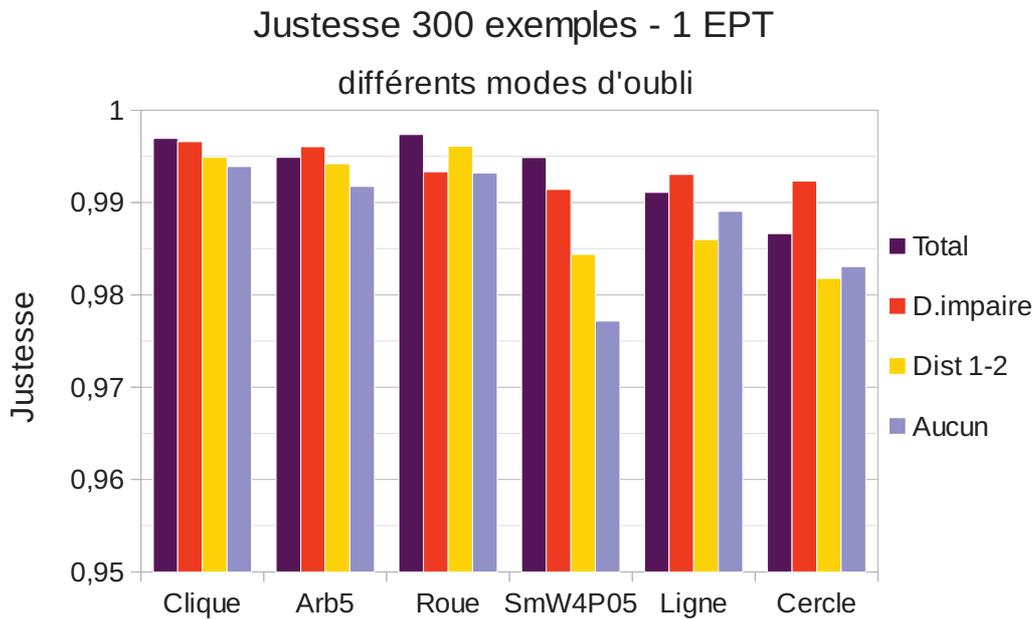


FIGURE 6.21 – Comparaison de la justesse d’apprentissage du problème M11, après 300 exemples dans différentes topologies de 40 agents, avec le protocole Waves et différentes formes d’oubli des exemples externes.



FIGURE 6.22 – Comparaison des temps d’exécution pour l’apprentissage du M11, après 300 exemples dans différentes topologies de 40 agents, avec le protocole Waves et différentes formes d’oubli des exemples externes.

Chapitre 7

Compléments et Discussion

Ce chapitre contient quelques remarques sur les résultats actuels ainsi que des discussions et prolongements possibles. Aucun nouveau résultat détaillé n'est présenté dans cette partie.

7.1 Relaxations de Waves

7.1.1 L'outil du tour par tour

La mise en place d'un tour par tour dans Waves permet de s'assurer que les exemples multiples d'un nouveau tour arrivent bien en même temps. Le but est d'assurer que les phases d'acquisition des exemples et d'apprentissage sont bien distinctes et de garantir que les hypothèses sont confrontées à tous les exemples du tour.

Un programme indépendant gère la simulation de l'environnement et donc la possibilité pour un agent de recevoir un nouvel exemple. C'est cet environnement qui doit savoir quand tous les agents ont fini de traiter leurs messages. En pratique, il faut que tous les agents soient dans l'état `Waiting` et qu'aucun d'eux n'ait encore de messages non traités, susceptibles de les faire changer d'état, pour autoriser le début d'un nouveau tour. Les agents ne peuvent passer de `Waiting` à `Apprentice` que suite à deux sortes de messages qui sont `new-Exemple` et `group-consistant`. On met donc en place un accusé de réception à la fin du protocole relatif à la lecture de ces deux messages. Les agents notifient à l'environnement lorsqu'ils changent d'état. L'accusé de réception de nouveaux exemples est attendu par l'environnement pour envoyer un `begin-turn` général. L'agent qui envoie un message `groupe-consistant` peut se contenter de demander un accusé de réception uniquement lorsqu'il

n'a plus d'autre hypothèse à traiter et qu'il compte changer d'état. Nous avons envisagé deux versions du tour par tour selon le traitement de l'accusé de réception de **groupe-consistant** :

- l'accusé de réception est envoyé à l'agent à l'origine du message. L'agent d'origine doit attendre toutes les réponses avant de notifier son passage en Waiting ou de traiter une nouvelle hypothèse reçue plus récemment.
- l'accusé de réception est envoyé à l'environnement. L'agent à l'origine du message **groupe-consistant** notifie à l'environnement le nombre d'accusés de réception attendus, mais passe immédiatement à l'état Waiting.

Le passage de la première solution à la deuxième est motivé par la libération des agents, de la gestion du tour par tour et de l'attente afin qu'il soient disponibles et actifs le plus possible. En pratique la transition a ralenti le protocole, rendant chaque tour un peu plus long. En effet l'attente des agents leur donne parfois le temps de recevoir plus de messages qui retirent des hypothèses à traiter et diminuent leur travail total. La deuxième solution est cependant conservée pour une raison à la fois conceptuelle et pratique : le tour par tour doit être indépendant du protocole Waves et on souhaite pouvoir le retirer.

7.1.2 Waves en continu

Le protocole Waves a été conçu pour pouvoir être analysé et proposer des garanties en tour par tour, mais également pour pouvoir fonctionner avec une arrivée continue d'exemples sans attendre la fin de chaque tour.

En théorie, la seule modification à apporter au protocole concerne la propriété $p1$: *Tout nouvel exemple (ou groupe d'exemples) reçu par le système doit être consistant avec les hypothèses personnelles de tous les agents ou donner lieu à une nouvelle hypothèse.* Les garanties de SMA-consistance en fin de tour ne sont maintenues que lorsqu'un tour se termine après le traitement de tous les messages et non par le début du suivant. Un agent n'est pas en mesure de savoir si le tour précédent s'est terminé lorsqu'il reçoit un nouvel exemple. Il n'est donc plus possible pour un agent de savoir s'il détient, dans ses hypothèses vérifiées, les hypothèses personnelles de tous les agents. Pour vérifier $p1$ un agent doit donc systématiquement proposer une nouvelle hypothèse à la réception d'un nouvel exemple, quitte à le faire dans des cas superflus.

En pratique, la date d'une hypothèse n'est plus un numéro de tour, mais l'heure de réception de l'exemple le plus récent dont elle tient compte. Elle est toujours utilisée pour déclencher l'oubli de toute hypothèse plus ancienne. Il peut arriver qu'un agent reçoive une hypothèse plus ancienne que celles dont il dispose, mais ce cas est déjà prévu dans le pré-traitement et l'hypothèse est simplement ignorée.

On peut s'attendre, en ne laissant pas les tours se terminer, à ce que la justesse obtenue au moment de l'arrivée de l'exemple $n+1$ soit moins bonne qu'en tour par tour. Le tour n n'ayant pas nécessairement le temps de se terminer, les hypothèses personnelles ne sont pas garanties SMA-consistantes.

Une expérience similaire avait été effectuée sur SMILE [BES07] et montrait qu'avec une arrivée régulière d'exemples, seul le début de l'apprentissage est pénalisé. Au début de l'apprentissage les révisions sont plus fréquentes et les tours ne se terminent pas. Quand l'activité diminue le SMA a le temps de finir ses tours et la justesse obtenue rattrape le cas du tour par tour.

On peut donc s'attendre avec Waves, en continu plutôt qu'en tour par tour, à ce que la justesse soit impactée sur le début de l'apprentissage, mais soit similaire sur la fin lorsque les révisions sont moins fréquentes. Par contre on peut également s'attendre à ce que cette justesse soit meilleure pour Waves qui bénéficie de la parallélisation que pour Variable-Root qui reste séquentiel.

7.1.3 Quelle hypothèse proposer ?

Première hypothèse du tour

Le choix actuel d'hypothèse à proposer en début de tour favorise le nombre de révisions avant le choix personnel de l'agent. Nous choisissons la première hypothèse vérifiée qui soit contredite par le nouvel exemple pour être révisée et proposée. On pourrait à l'inverse choisir la première hypothèse qui ne nécessite pas de révision ou encore systématiquement prendre l'hypothèse personnelle de l'agent, qu'elle nécessite une révision ou non. Comme il n'est plus possible, en continu, d'éviter de faire vérifier une hypothèse par l'ensemble du réseau à chaque tour, il n'est pas nécessaire de chercher une hypothèse contredite dans l'ensemble des hypothèses vérifiées.

Re-proposition d'une hypothèse

On peut envisager que les agents ne souhaitent pas voir leur hypothèse personnelle simplement disparaître au profit d'une plus récente à chaque nouveau tour. L'hypothèse qu'ils souhaitent conserver doit alors être de nouveau

vérifiée pour avoir une garantie de SMA-consistance. Il est possible, à la réception d'une hypothèse plus récente de conserver, une ou plusieurs des hypothèses en mémoire, mais il faut alors les placer dans les hypothèses à traiter avec la date de l'hypothèse plus récente et de nouveaux identifiants. Ces hypothèses devront alors être proposée de nouveau pour être vérifiées par l'ensemble du réseau.

7.1.4 Relaxations

Groupe-consistance mais à profondeur limitée

Plus on s'approche d'une arrivée rapide de nouveaux exemples ou d'une taille de réseau conséquente, plus on peut souhaiter faire des compromis sur la qualité des hypothèses recherchée pour accélérer le protocole. On peut alors envisager de relâcher la SMA-consistance en plus du tour par tour. Par exemple on peut choisir de fixer une limite de distance pour la propagation (et vérification) des hypothèses ainsi que des exemples. On recherche alors toujours une groupe-consistance, mais à profondeur limitée.

Propagation d'opinion et adoption modérée

On peut envisager que les agents aient des préférences et que leur conception du monde soit plus proche du domaine de l'opinion que de celui du savoir. Ils décident de la représentation du monde qu'ils utilisent comme référence, non seulement par des raisonnements logiques à partir de faits qu'ils récoltent avec l'aide de leur proches, mais aussi par l'influence du réseau auquel ils appartiennent. On peut envisager de représenter l'adoption d'hypothèse par un modèle mathématique de propagation d'opinion. Avec Waves, le traitement et la propagation des hypothèses reçues par messages de SMA-consistance pourraient n'avoir lieu que dans certaines conditions plus subtiles qu'une limite de distance. On pourrait par exemple s'inspirer d'une approche par essais particuliers comme Maleszka [Mal17] l'a fait pour un modèle multi-agents disposant de connaissances hétérogènes.

Remise en cause de la consistance

En considérant que l'on ne tient plus compte de l'ensemble des agents pour valider une hypothèse et que celle-ci n'est plus transmise à l'ensemble du réseau, on peut alors aller jusqu'à remettre en cause le choix de l'apprentissage exact. Il est en effet possible que certaines informations, agents ou moyen de communication ne soient pas dignes de confiance et que l'on souhaite en tenir compte dans le protocole. De manière générale, Waves, comme

SMILE, est peu impacté par la présence d'exemples erronés en faible proportions. Il n'est cependant pas conçu ni pour les détecter ni pour en tenir compte différemment dans son apprentissage.

7.2 Effets d'une confiance non absolue

En retirant le tour par tour ou en demandant une restitution pendant un tour sans en attendre la fin, on perd la garantie de SMA-consistance des agents. On peut vouloir économiser du temps ou des messages et rechercher un bon résultat sans qu'il soit nécessairement de l'ordre de la SMA-consistance pour tout agent. Une modification du protocole Waves sera alors à envisager en déterminant préalablement quelles relaxations sont envisageables sans pertes conséquentes de justesse.

Les problèmes de confiances entre agents et dans le réseau posent également la question de la nécessité de la SMA-consistance.

7.2.1 Incohérence de comportement, mais cohérence de protocole

Afin d'alléger le protocole, on se permet d'éviter des actions redondantes des agents. Cela représente une structure dans laquelle les agents ont particulièrement confiance en les activités des autres. La cohérence du protocole à l'échelle du réseau permet de garantir la SMA-consistance des hypothèses vérifiées à terme, mais le comportement individuel des agents peut être vu comme incohérent.

- En particulier un agent ne vérifie pas lors de la réception d'un contre exemple si ses autres hypothèses vérifiées sont toujours consistantes.
- Un agent ne vérifie pas non plus la consistance d'une hypothèse reçue dans un message `group-consistant(H_c)` avant de la proposer. En raison de la liste d'attente des hypothèses à traiter, il est possible avec Waves actuel que l'agent possède un contre exemple à l'hypothèse H_c lorsqu'elle devient l'hypothèse actuelle. Il a pu acquérir ce contre exemple lors de la révision d'une précédente hypothèse. Ce cas, peu probable, n'est pas problématique à l'échelle globale du protocole. Dans tous les cas l'hypothèse étant confrontée à tous les agents, sa non-SMA-consistance sera détectée de façon certaine par l'agent qui dispose de ce contre-exemple comme exemple interne.

En plus de la problématique actuelle de la cohérence de comportement d'un agent, si des problèmes de fiabilité ou de sûreté du réseau apparaissent il faudra apporter les modifications suivantes à Waves :

- vérifier la consistance d'une hypothèse reçue par un message `group-consistant` avec ses propres exemples avant de la proposer.
- à la réception d'un contre exemple, s'assurer de la consistance des hypothèses vérifiées avec cet exemple et les réviser si nécessaire.
- Limiter l'oubli des exemples externes s'il y a un risque de perdre contact avec leur origine.

7.2.2 Envisager une τ -acceptabilité

Principe original Dans leur article, Santiago Ontañón et Enric Plaza [OP14], utilisent un coefficient pour évaluer *la confiance* portée à une hypothèse.

La confiance (en une hypothèse) est définie comme le (nombre exemples positif couverts +1) divisé par le (nombre total d'exemples couverts + 2)

Le 1 du numérateur et 2 du dénominateur correspondent à la correction de Laplace et permet d'avoir peu confiance en une hypothèse couvrant peu d'exemples au total.

Adaptation à Waves Dans Waves, on pourrait remplacer le critère de SMA-consistance par un critère plus souple de *coefficient d'acceptabilité* inspiré de la confiance. Afin de tenir compte équitablement des exemples positifs et négatifs on propose le coefficient : (nombre d'exemples bien classés +1) sur (nombre d'exemples totaux +2). Une hypothèse doit avoir un coefficient d'acceptabilité atteignant une valeur **seuil** τ pour être acceptée. Le choix entre plusieurs hypothèses peut alors se faire avec une fonction de score dépendant du problème d'apprentissage traité.

Le Coefficient d'acceptabilité On peut considérer trois niveaux pour ce coefficient, celui de l'agent, celui d'un groupe et celui du SMA).

Au niveau d'un agent on peut comptabiliser tous les exemples qu'il a en mémoire, mais on préférera compter uniquement les exemples internes. La valeur du coefficient obtenue à l'échelle d'un seul agent est alors la valeur de *perso-acceptabilité*. Un agent qui propose une hypothèse peut obtenir une réponse avec un nouveau message dont le sens est : "voici le nombre de contre-exemples internes que je possède et le nombre d'exemples internes totaux que

je possède". En faisant la somme des réponses de tous les agents, on peut obtenir le nombre d'exemples consistants avec une hypothèse ainsi que le nombre total d'exemple différents. Le rapport de ces deux nombres donne un coefficient d'acceptabilité d'une hypothèse au niveau du groupe ou du SMA si on y ajoute un mécanisme de propagation. On définit ainsi les valeurs de groupe-acceptabilité et *SMA-acceptabilité*.

Fonction de score On peut vouloir donner de l'importance aux hypothèses ou aux coefficients de τ -acceptabilité des autres agents selon différents critères. Ces critères sont plus ou moins importants selon la situation. Voici quelques critères et les raisons pour lesquelles ils peuvent être important à favoriser :

- l'**agent d'origine** peut être important d'une part dans le cas d'agents non pertinents disposant de peu d'exemples et d'autre part s'il y a présence d'agents malveillants ou défectueux
- La **taille** d'une hypothèse
- le **coefficient d'acceptabilité** dans lequel on peut éventuellement pondérer les contre-exemples avec le **temps** ou la **distance** si le concept à apprendre varie dans le temps ou l'espace.

7.2.3 Approche perso-consistante, SMA-acceptable

Problème d'apprentissage modélisé

C'est une approche qui représente une confiance totale d'un agent en ses propres données et expériences. On considère ici qu'un agent ne fait pas d'erreur, mais qu'il n'a pas une confiance totale dans le travail des autres. Les raisons peuvent être la présence d'**agents malicieux ou défectueux** ou la possibilité d'une **dérive de concept selon le lieu** ou encore la volonté d'exprimer une **variante locale** simplifiée d'un concept commun dans le cas d'une répartition non uniforme des exemples.

Interactions entre agents

Un agent forme son hypothèse à partir de ses exemples internes il demande alors une évaluation de cette hypothèse au niveau SMA. Il construit une fonction de score basée sur le coefficient d'acceptabilité en fonction des réponses de chaque agents. Si elle atteint un certain seuil τ il la conserve sinon il utilise des contre-exemples externes pour l'améliorer. Il tente d'être

consistant avec les exemples qu'il estime avoir le plus de poids s'il y a pondération (origine, distance), mais pas nécessairement tous.

7.2.4 Approche perso-acceptable, SMA-acceptable

Problème d'apprentissage modélisé

On considère cette fois la possibilité de **bruit**, c'est-à-dire d'exemples erronés, ou encore de **dérive de concept dans le temps**. Les données acquises par l'agent ne sont donc pas plus sûres que celles qu'il reçoit des autres agents. On peut également vouloir envisager qu'un critère devienne plus important que la consistance. Il faut alors former un compromis entre ce critère, la taille d'hypothèse par exemple et la proximité de la consistance. On s'approche alors de l'apprentissage agnostique en voulant apprendre au mieux plutôt que parfaitement. Dans ce cas il n'y a pas de raison d'exiger un apprentissage exact de la part de l'agent formant la nouvelle hypothèse, mais uniquement un niveau d'acceptabilité comme pour les autres agents.

Interactions entre agents

Dans cette approche, on choisit de contraindre l'hypothèse que l'on veut former en priorité sur un autre critère que sa consistance. Une fonction de score peut être nécessaire pour estimer s'il faut déclencher une révision ou non. Le processus d'apprentissage actuel, *Imind2*, recherche la consistance. Il faut alors également chercher un mécanisme de formation et révision d'hypothèse alternatif.

7.2.5 Apprentissage réaliste

Les problèmes d'apprentissages peuvent mélanger le bruit constitué d'erreurs de mesure accidentelles et d'autres phénomènes de localité ou de confiance entre agents. Il peut devenir nécessaire alors d'utiliser d'une τ acceptabilité pondérée ou encore de remettre en cause le processus d'apprentissage.

7.3 Prolongements et analyses complémentaires

7.3.1 Résultats de Waves

Il serait intéressant de poursuivre l'analyse de Waves selon plusieurs critères :

- l'effet de l'augmentation du nombre d'agents.
- la comparaison de SMILE en réseau sans propagation à Waves selon d'autres ressources que le nombre d'exemples. Pour une même justesse atteinte on peut considérer que les exemples ne sont pas limitants et chercher à comparer le temps de calcul de chaque agent, le nombre de messages ou encore la taille mémoire nécessaire.
- tenir compte de la robustesse des réseaux dans l'évaluation de leur efficacité d'apprentissage. Il s'agit ici de considérer l'impact du retrait d'un agent ou d'une arête sur l'apprentissage en moyenne et dans le pire cas.
- l'intérêt éventuel d'un protocole à propagation limité soit pour simplifier le changement d'échelle soit pour simplifier les hypothèses selon des spécificités locales.
- observer les déplacements réels des exemples au cours de l'apprentissage.

7.3.2 Réseaux de structure dynamique

Les réseaux étudiés jusqu'à présents sont tous fixes. Il serait intéressant de considérer des réseaux avec des liens dynamiques tels que les réseaux pair-à-pair. Le protocole Waves ne devrait pas être affecté, tant que les agents sont informés des connexions et déconnexions de leurs voisins. Il faut bien sûr adapter les compteurs de réponses attendues lors d'un départ et re-proposer les hypothèses vérifiées à un voisin nouveau venu. Le nouveau venu doit également proposer ses hypothèses. La SMA-consistance des hypothèses pourra ne plus être garantie si le réseau vient à ne plus être connexe.

7.3.3 Répartition non uniforme des exemples

Deux expérimentations sont envisagées concernant la répartition non uniforme des exemples.

Une première consiste à observer l'apprentissage réalisé par un SMA selon les différents protocoles SMILE, Variable-Root et Waves lorsque les agents cherchent tous à apprendre le même concept, mais que la répartition des exemples reçus par les agents n'est plus uniforme.

Une seconde étude revient à limiter la propagation des exemples et des hypothèses à une certaine distance. Il est alors intéressant d'étudier comment les agents apprennent selon leur localité. La répartition des exemples

doit alors refléter un même concept général, mais qui ne s'exprime pas en totalité partout. Les agents peuvent se contenter d'une représentation simplifiée du concept pour prédire les exemples locaux. Des concepts relatifs à la météo dans différents pays peuvent illustrer ce principe de vérité générale qui s'exprime partiellement différemment selon le lieu.

Chapitre 8

Conclusion

Une première étude apportée par ce document concerne la limitation de la propagation de l'information dans un graphe. Les diverses structures de graphe ainsi que le vecteur d'information (exemple ou hypothèse) n'ont pas la même influence sur l'efficacité de l'apprentissage lorsqu'on limite les révisions globales à des révisions de groupes. Ce principe de limiter les agents à des révisions de groupes, en ne leur permettant de contacter que leur voisinage le plus proche, pourra être réutilisé pour aborder différents problèmes d'apprentissage. Ces situations peuvent être la mise en place d'un moyen de relâcher la contrainte de SMA-consistance ou encore une solution pour s'adapter à une répartition non uniforme des exemples. L'apprentissage réseau limité au voisinage, détaillé dans ce document, pourra alors servir de référence pour de nouvelles situations comme celles sus-citées. Il apporte d'ores et déjà un éclairage sur des phénomènes d'apprentissages collectifs.

Dans une seconde étude, un nouveau protocole est présenté : le protocole Waves. Le protocole Waves permet de résoudre en même temps deux difficultés qui n'avaient jusqu'alors été dépassées que séparément dans le cadre de l'apprentissage collectif. La première difficulté est la simultanéité des communications. La seconde est la limitation des communications possibles, ce que l'on caractérise par une structure de réseau. La simultanéité de ces deux caractéristiques doit également permettre l'apprentissage de plusieurs agents dans un même SMA, pour atteindre un fonctionnement dynamique en parallèle.

Waves est donc un protocole dynamique fonctionnant en parallèle. Il offre de plus des garanties de terminaison et de consistance globale.

Le protocole Waves a été amélioré et optimisé pour en limiter les redondances. Il économise ainsi des messages, des exemples en mémoire et surtout

des calculs pour une exécution finale plus rapide.

Un compromis, entre la justesse recherchée et le temps disponible, peut être atteint au moyen de différents modes d'oubli des exemples externes.

Plusieurs prolongements ont été proposés et l'étude de certains d'entre eux pourront venir en complément de cette thèse sur l'apprentissage artificiel collectif dans ses aspects dynamiques et structurels.

Bibliographie

- [Aly05] Mohamed Aly. Survey on multiclass classification methods. 2005.
- [Ang88] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4) :319–342, 1988.
- [Ang98] Eric Angel. *La rugosité des paysages : une théorie pour la difficulté des problèmes d’optimisation combinatoire relativement aux meta-heuristiques*. PhD thesis, 1998. Thèse de doctorat dirigée par Fayard, Didier Terre, ocean, espace Paris 11 1998.
- [Ang04] Dana Angluin. Queries revisited. *Theor. Comput. Sci.*, 313(2) :175–194, 2004.
- [ASS01] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary : A unifying approach for margin classifiers. *J. Mach. Learn. Res.*, 1 :113–141, September 2001.
- [BD11] Gaston Bonnier and Robert Douin. *Flore complète illustrée en couleurs de France : Suisse et Belgique (comprenant la plupart des plantes d’Europe)*, volume 1. Delachaux et Niestlé, 1911.
- [BDGP90] R. Beckers, J. L. Deneubourg, S. Goss, and J. M. Pasteels. Collective decision making through food recruitment. *Insectes Sociaux*, 37(3) :258–267, Sep 1990.
- [BEMS08] Gouvain Bourgne, Amal El Fallah Seghrouchni, Nicolas Maudet, and Henry Soldano. Multiagent incremental learning in networks. In The Duy Bui, Tuong Vinh Ho, and Quang-Thuy Ha, editors, *Pacific Rim International Conference on Multi-Agents (PRIMA)*, volume 5357 of *Lecture Notes in Computer Science*, pages 109–120, Hanoi, Vietnam, December 2008. Springer.

- [BES07] Gauvain Bourgne, Amal El Fallah Seghrouchni, and Henry Soldano. Smile : Sound multi-agent incremental learning ;-). In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 164–171, Honolulu, Hawaiï, May 14–18 2007. ACM.
- [BES09] Gauvain Bourgne, Amal El Fallah Seghrouchni, and Henry Soldano. Learning in a fixed or evolving network of agents. In *IEEE / WIC / ACM International Conference on Intelligent Agent Technology*, pages 549–556, Milano, Italy, 2009. IEEE Computer Society.
- [Bet12] Gregor Betz. *Debate Dynamics : How Controversy Improves Our Beliefs*. Springer Publishing Company, Incorporated, 2012.
- [BFSO84] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.
- [Bis95] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [Bis06] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.
- [Bla88] Agnès Blaye. *Confrontation socio-cognitive et résolution de problème*. PhD thesis, Université de Provence, 1988.
- [Bre96] Leo Breiman. Bagging predictors. *Machine learning*, 24(2) :123–140, 1996.
- [Bro11] Gavin Brown. Ensemble learning. In *Encyclopedia of Machine Learning*, pages 312–320. Springer, 2011.
- [BS97] Marko Balabanović and Yoav Shoham. Fab : Content-based, collaborative recommendation. *Commun. ACM*, 40(3) :66–72, March 1997.
- [BS13] Kanishka Bhaduri and Marco Stolpe. Distributed data mining in sensor networks., 2013.
- [BS14] Dominique Bouthinon and Henry Soldano. Learning first order rules from ambiguous examples. In Goerge A. Papadopoulos, editor, *ICTAI*, page to appear, Limassol, November 2014. IEEE.

- [BSE10] Gauvain Bourgne, Henry Soldano, and Amal El Fallah Seghrouchni. Learning better together. In Helder Coelho, Rudi Studer, and Michael Wooldridge, editors, *ECAI*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 85–90. IOS Press, 2010.
- [BSvL12] Sander C.J. Bakkes, Pieter H.M. Spronck, and Giel van Lankveld. Player behavioural modelling for video games. *Entertainment Computing*, 3(3) :71 – 79, 2012. Games and AI.
- [Cam91] Roberto Camagni. *Innovation networks*. John Wiley & Sons, Inc., 1991.
- [Cap07] Roberta Capello. Spatial transfer of knowledge in high technology milieux : Learning versus collective learning processes (volume 33, number 4, 1999). *Regional Studies*, 41(sup1) :S161–S173, 2007.
- [Cel10] Ò. Celma. *Music Recommendation and Discovery : The Long Tail, Long Fail, and Long Play in the Digital Music Space*. SpringerLink : Springer e-Books. Springer Berlin Heidelberg, 2010.
- [CL87] C. J. Colbourn and P. H. Light. Social interaction and learning using micro-prolog. *Journal of Computer Assisted Learning*, 3(3) :130–140, 1987.
- [CM11] Antoine Cornuéjols and Laurent Miclet. *Apprentissage artificiel : concepts et algorithmes*. Editions Eyrolles, 2011.
- [COH09] Jongeun Choi, Songhwa Oh, and Roberto Horowitz. Distributed learning and cooperative control for multi-agent systems. *Automatica*, 45(12) :2802 – 2814, 2009.
- [Cui16] Jin Cui. *Data Aggregation in Wireless Sensor Networks*. Theses, INSA Lyon, June 2016.
- [DB95] T. G. Dietterich and G. Bakiri. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *eprint arXiv :cs/9501101*, December 1995.
- [DBBO95] Pierre Dillenbourg, Michael J Baker, Agnes Blaye, and Claire O’Malley. The evolution of research on collaborative learning, 1995.

- [DBG⁺06] Souptik Datta, Kanishka Bhaduri, Chris Giannella, Ran Wolff, and Hillol Kargupta. Distributed data mining in peer-to-peer networks. *IEEE Internet Computing*, 10(4) :18–26, 2006.
- [DE95] WHE Davies and Peter Edwards. Distributed learning : An agent-based approach to data-mining. 1995.
- [DG01] Carlotta Domeniconi and Dimitrios Gunopulos. Incremental support vector machine construction. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 589–592. IEEE, 2001.
- [DH73] Richard O Duda and Peter E Hart. Pattern recognition and scene analysis, 1973.
- [Dil99a] P. Dillenbourg. *Collaborative Learning : Cognitive and Computational Approaches*. Advances in learning and instruction series. Elsevier Science & Technology Books, 1999.
- [Dil99b] Pierre Dillenbourg. What do you mean by collaborative learning?, 1999.
- [DK82] Pierre A Devijver and Josef Kittler. *Pattern recognition : A statistical approach*. Prentice hall, 1982.
- [DM84] W. Doise and G. Mugny. *The Social Development of the Intellect*. International Series of Monographs in the Science of the Solid State. Elsevier Science Limited, 1984.
- [DZM⁺09] Haimonti Dutta, Xianshu Zhu, Tushar Mahule, Hillol Kargupta, Kirk D. Borne, Codrina Lauth, Florian Holz, and Gerhard Heyer. Taglearner : A P2P classifier learning system from collaboratively tagged text documents. In *ICDM Workshops 2009, IEEE International Conference on Data Mining Workshops, Miami, Florida, USA, 6 December 2009*, pages 495–500, 2009.
- [EM89] Shimon Even and Burkhard Monien. On the number of rounds necessary to disseminate information. In *Proceedings of the first annual ACM symposium on Parallel algorithms and architectures*, pages 318–327. ACM, 1989.
- [EM04] Saher Esmeir and Shaul Markovitch. Lookahead-based algorithms for anytime induction of decision trees. In *ICML’04*, pages 257–264. Morgan Kaufmann, 2004.

- [ETpLPG13] Z. Erkin, J. R. Troncoso-pastoriza, R. L. Legendijk, and F. Perez-Gonzalez. Privacy-preserving data aggregation in smart metering systems : an overview. *IEEE Signal Processing Magazine*, 30(2) :75–86, March 2013.
- [Fer89] Jacques Ferber. *Objets et agents : une étude des structures de représentation et de communications en intelligence artificielle*. PhD thesis, 1989. Thèse de doctorat dirigée par Perrot, Jean-François Sciences appliquées Paris 6 1989.
- [FLS07] K. W. Fan, S. Liu, and P. Sinha. Structure-free data aggregation in sensor networks. *IEEE Transactions on Mobile Computing*, 6(8) :929–942, Aug 2007.
- [Gas91] Les Gasser. Social conceptions of knowledge and action : Dai foundations and open systems semantics. *Artificial Intelligence*, 47(1) :107 – 138, 1991.
- [GCMH⁺10] Álvaro Gutiérrez, Alexandre Campo, Félix Monasterio-Huelin, Luis Magdalena, and Marco Dorigo. Collective decision-making based on social odometry. *Neural Computing and Applications*, 19(6) :807–823, Sep 2010.
- [GHEFSS05] Alejandro Guerra-Hernández, Amal El Fallah-Seghrouchni, and Henry Soldano. *Learning in BDI Multi-agent Systems*, pages 218–233. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [HB05] Thomas Hofmann and Justin Basilico. Collaborative machine learning. In Matthias Hemmje, Claudia Niederée, and Thomas Risse, editors, *From Integrated Publication and Information Systems to Information and Knowledge Environments : Essays Dedicated to Erich J. Neuhold on the Occasion of His 65th Birthday*, pages 173–182. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [HCB02] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless micro-sensor networks. *IEEE Transactions on Wireless Communications*, 1(4) :660–670, Oct 2002.
- [Hen97] M’hammed Henniche. Apprendre incrémentalement dans un l’espace des généralisations maximales spécifiquement d’ins-

- tances. In *Journées Francophones d'Apprentissage, Roscoff*, pages 125–135, 1997.
- [HT98] Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. In *Advances in neural information processing systems*, pages 507–513, 1998.
- [IJL⁺09] Shadi Ibrahim, Hai Jin, Lu Lu, Li Qi, Song Wu, and Xuanhua Shi. Evaluating mapreduce on virtual machines : The hadoop case. In Martin Gilje Jaatun, Gansen Zhao, and Chunming Rong, editors, *Cloud Computing : First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings*, pages 519–528. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [Jar05] Isabelle Jars. *Contribution des Sciences Sociales dans le domaine de l'Intelligence Artificielle Distribuée : ALONE, un modèle hybride d'agent apprenant*. PhD thesis, Université Claude Bernard-Lyon I, 2005.
- [JM96] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile computing*, pages 153–181, 1996.
- [KBHJ99] Hillol Kargupta, Daryl Hershberger Byung-Hoon, and Erik Johnson. Collective data mining : A new perspective toward distributed data analysis. In *Advances in distributed and parallel knowledge discovery*. Citeseer, 1999.
- [KD15] Mukesh Kumar and Kamlesh Dutta. *A Survey of Security Concerns in Various Data Aggregation Techniques in Wireless Sensor Networks*, pages 1–15. Springer India, New Delhi, 2015.
- [KEW02] L. Krishnamachari, D. Estrin, and S. Wicker. The impact of data aggregation in wireless sensor networks. In *Proceedings 22nd International Conference on Distributed Computing Systems Workshops*, pages 575–578, 2002.
- [KGP⁺10] Hillol Kargupta, Michael Gilligan, Vasundhara Puttagunta, Kakali Sarkar, Martin Klein, Nick Lenzi, and Derek Johnson. Minefleet® : The vehicle data stream mining system for ubiquitous environments. In *Ubiquitous Knowledge Discovery - Challenges, Techniques, Applications*, pages 235–254. 2010.

- [KL05] N. Kimura and S. Latifi. A survey on data compression in wireless sensor networks. In *International Conference on Information Technology : Coding and Computing (ITCC'05) - Volume II*, volume 2, pages 8–13 Vol. 2, April 2005.
- [KSSV00] Richard Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. Randomized rumor spreading. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 565–574. IEEE, 2000.
- [Kuh72] Deanna Kuhn. Mechanisms of change in the development of cognitive structures. *Child Development*, 43(3) :833–844, 1972.
- [Kun] Ludmila I Kuncheva. Classifier ensembles for changing environments.
- [KW99] David Keeble and Frank Wilkinson. Collective learning and knowledge development in the evolution of regional clusters of high technology smes in europe. *Regional Studies*, 33(4) :295–303, 1999.
- [Lan95] Ken Lang. Newsweeder : Learning to filter netnews. In Armand Frieditis and Stuart J. Russell, editors, *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, pages 331–339. Morgan Kaufmann, 1995.
- [LBHS98] Rikard Larsson, Lars Bengtsson, Kristina Henriksson, and Judith Sparks. The interorganizational learning dilemma : Collective knowledge development in strategic alliances. *Organization Science*, 9(3) :285–305, 1998.
- [LL99] Clive Lawson and Edward Lorenz. Collective learning, tacit knowledge and regional innovative capacity. *Regional Studies*, 33(4) :305–317, 1999.
- [LSY03] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations : Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1) :76–80, January 2003.
- [LW⁺02] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3) :18–22, 2002.

- [LWHL14] Hongxing Li, Chuan Wu, Qiang-Sheng Hua, and Francis C.M. Lau. Latency-minimizing data aggregation in wireless sensor networks under physical interference model. *Ad Hoc Networks*, 12(Supplement C) :52 – 68, 2014. (1) SCENarios for ad hoc Network Evaluation Studies (SCENES); (2) Modeling, Analysis and Simulation of Wireless and Mobile Systems and (3) Algorithmic Aspects of Ad-Hoc Networks.
- [LWT05] Chong Liu, Kui Wu, and Min Tsao. Energy efficient information collection with the arima model in wireless sensor networks. In *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, volume 5, pages 5 pp.–2474, Dec 2005.
- [Mal17] Marcin Maleszka. Particle swarm of agents for heterogenous knowledge integration. In Nguyen et al. [NPJ⁺17], pages 54–62.
- [MBD⁺10] Tushar Mahule, Kirk D. Borne, Sandipan Dey, Sugandha Arora, and Hillol Kargupta. PADMINI : A peer-to-peer distributed astronomy data mining system and a case study. In *Proceedings of the 2010 Conference on Intelligent Data Understanding, CIDU 2010, October 5-6, 2010, Mountain View, California, USA*, pages 243–257, 2010.
- [Mel98] Gabor Melli. *A Lazy Model-Based Approach to On-Line Classification*. PhD thesis, University of British Columbia, 1998.
- [MFHH02] Samuel Madden, Michael J Franklin, Joseph M Hellerstein, and Wei Hong. Tag : A tiny aggregation service for ad-hoc sensor networks. *ACM SIGOPS Operating Systems Review*, 36(SI) :131–146, 2002.
- [MPF01] E. Mallon, S. Pratt, and N. Franks. Individual and collective decision-making during nest site selection by the ant *leptothorax albipennis*. *Behavioral Ecology and Sociobiology*, 50(4) :352–359, Sep 2001.
- [MW12] Winter Mason and Duncan J. Watts. Collaborative learning in networks. *Proceedings of the National Academy of Sciences*, 109(3) :764–769, January 2012.
- [NPJ⁺17] Ngoc Thanh Nguyen, George A. Papadopoulos, Piotr Jedrzejowicz, Bogdan Trawinski, and Gottfried Vossen, editors. *Computational Collective Intelligence - 9th International Conference*,

ICCCI 2017, Nicosia, Cyprus, September 27-29, 2017, Proceedings, Part I, volume 10448 of *Lecture Notes in Computer Science*. Springer, 2017.

- [OP10] Santiago Ontañón and Enric Plaza. Multiagent inductive learning : an argumentation-based approach. In Johannes Fürnkranz and Thorsten Joachims, editors, *ICML*, pages 839–846. Omnipress, 2010.
- [OP14] Santiago Ontañón and Enric Plaza. Coordinated inductive learning using argumentation-based communication. *Autonomous Agents and Multi-Agent Systems*, 29 :266–304, 2014.
- [PBP10] Krishna P. N. Puttaswamy, Ranjita Bhagwan, and Venkata N. Padmanabhan. Anonymgator : Privacy and integrity preserving data aggregation. In Indranil Gupta and Cecilia Mascolo, editors, *Middleware 2010 : ACM/IFIP/USENIX 11th International Middleware Conference, Bangalore, India, November 29 - December 3, 2010. Proceedings*, pages 85–106. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann Publishers, 1988.
- [PH96] Foster J Provost and Daniel N Hennessy. Scaling up : Distributed machine learning with cooperation. In *AAAI/IAAI, Vol. 1*, pages 74–79, 1996.
- [Pia23] Jean Piaget. *La langage et la pensée chez l’enfant : Études sur la logique de l’enfant*. 1923.
- [PZ01] Patrice Perny and Jean-Daniel Zucker. Preference-based Search and Machine Learning for Collaborative Filtering : the "Film-Conseil" Movie Recommender System. *Revue I3 - Information Interaction Intelligence*, 1(1) :1–40, 2001.
- [QR89] J Ross Quinlan and Ronald L Rivest. Inferring decision trees using the minimum description lenght principle. *Information and computation*, 80(3) :227–248, 1989.
- [RG91] Anand S. Rao and Michael P. Georgeff. Modeling rational agents within a BDI-architecture. In James Allen, Richard

- Fikes, and Erik Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 473–484. Morgan Kaufmann publishers Inc. : San Mateo, CA, USA, 1991.
- [RIS⁺94] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens : An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94*, pages 175–186, New York, NY, USA, 1994. ACM.
- [Ris01] Irina Rish. An empirical study of the naïve bayes classifier. 3, 01 2001.
- [RK04] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of machine learning research*, 5(Jan) :101–141, 2004.
- [Ros92] Jeremy Roschelle. Learning by collaborating : Convergent conceptual change. *Journal of the Learning Sciences*, 2(3) :235–276, 1992.
- [RR88] Barbara Radziszewska and Barbara Rogoff. Influence of adult and peer collaborators on children’s planning skills. 24 :840–848, 11 1988.
- [RSBR14] Christophe Rodrigues, Henry Soldano, Gauvain Bourgne, and Céline Rouveirol. Multi agent learning of relational action models. In Torsten Schaub, Gerhard Friedrich, and Barry O’Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 1087–1088. IOS Press, 2014.
- [RT95] Jeremy Roschelle and Stephanie D. Teasley. The construction of shared knowledge in collaborative problem solving. In Claire O’Malley, editor, *Computer Supported Collaborative Learning*, pages 69–97. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995.
- [SB98] Richard S Sutton and Andrew G Barto. *Reinforcement learning : An introduction*, volume 1. MIT press Cambridge, 1998.

- [Sch03] Robert E Schapire. The boosting approach to machine learning : An overview. In *Nonlinear estimation and classification*, pages 149–171. Springer, 2003.
- [SCS91] Thomas D. Seeley, Scott Camazine, and James Sneyd. Collective decision-making in honey bees : how colonies choose among nectar sources. *Behavioral Ecology and Sociobiology*, 28(4) :277–290, Apr 1991.
- [Set12] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1) :1–114, 2012.
- [SFKD99] F Saffre, R Furey, B Krafft, and J.L Deneubourg. Collective decision-making in social spiders : Dragline-mediated amplification process acts as a recruitment mechanism. *Journal of Theoretical Biology*, 198(4) :507 – 517, 1999.
- [SG02] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec) :583–617, 2002.
- [SHM⁺16] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search : Nature : Nature research. *Nature*, 529 :484–489, January 2016.
- [SKPP05] Jeong-Hun Shin, Jaesub Kim, Keuntae Park, and Daeyeon Park. Railroad : Virtual infrastructure for data dissemination in wireless sensor networks. In *Proceedings of the 2Nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, PE-WASUN '05, pages 168–174, New York, NY, USA, 2005. ACM.
- [SM95] Upendra Shardanand and Pattie Maes. Social information filtering : Algorithms for automating “word of mouth”. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

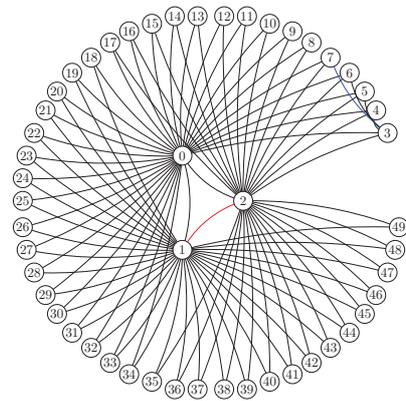
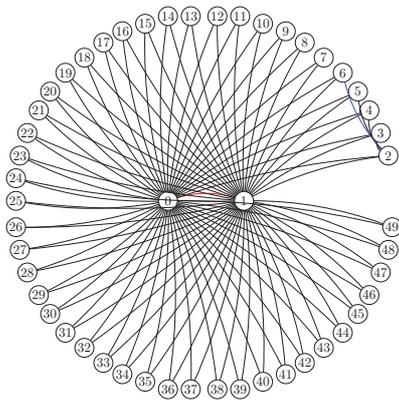
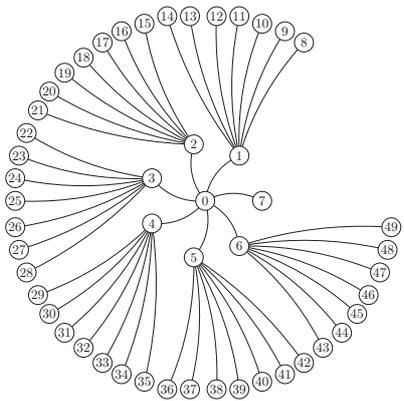
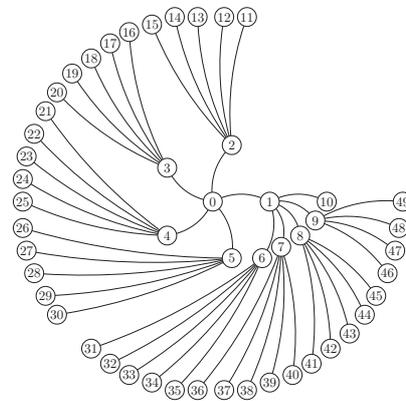
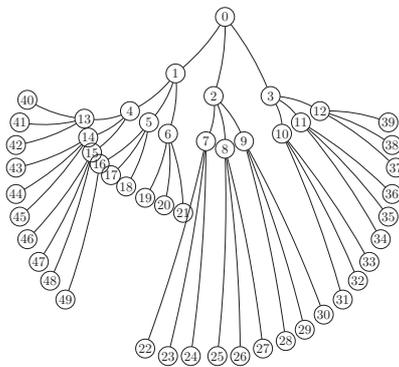
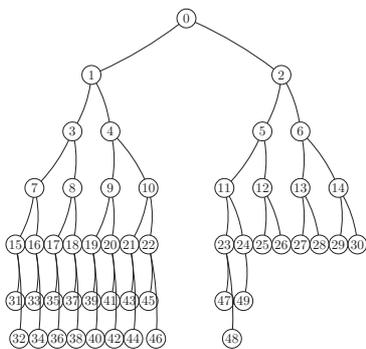
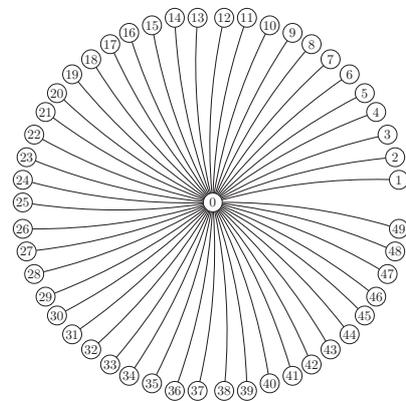
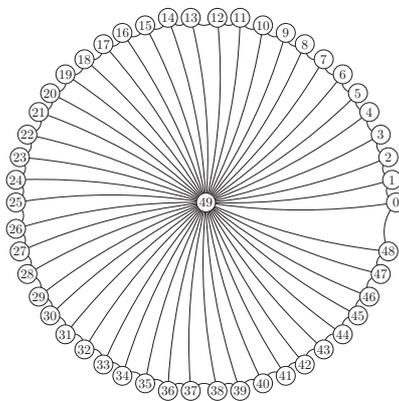
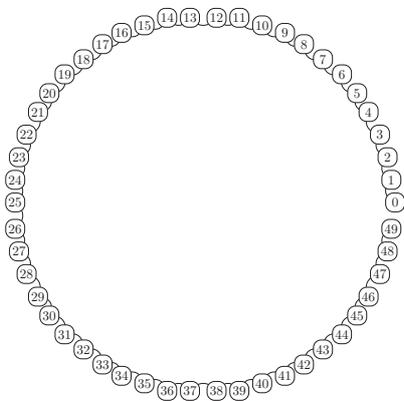
- [SMC⁺17] Jérémie Sublime, Basarab Matei, Guénaél Cabanes, Nistor Grozavu, Younès Bennani, and Antoine Cornuéjols. Entropy based probabilistic collaborative clustering. *Pattern Recognition*, 72(Supplement C) :144 – 157, 2017.
- [SVS⁺12] Thomas D. Seeley, P. Kirk Visscher, Thomas Schlegel, Patrick M. Hogan, Nigel R. Franks, and James A. R. Marshall. Stop signals provide cross inhibition in collective decision-making by honeybee swarms. *Science*, 335(6064) :108–111, 2012.
- [TD84] David Trowbridge and Robin Durnin. Results from an investigation of groups working at the computer. 1984.
- [Vap95] Vladimir N Vapnik. The nature of statistical learning theory. 1995.
- [VBS17a] Lise-Marie Veillon, Gauvain Bourgne, and Henry Soldano. Effect of network topology on neighbourhood-aided collective learning. In Nguyen et al. [NPJ⁺17], pages 202–211.
- [VBS17b] Lise-Marie Veillon, Gauvain Bourgne, and Henry Soldano. Waves : a model of collective learning. In Amit P. Sheth, Axel Ngonga, Yin Wang, Elizabeth Chang, Dominik Slezak, Bogdan Franczyk, Rainer Alt, Xiaohui Tao, and Rainer Unland, editors, *Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, August 23-26, 2017*, pages 314–321. ACM, 2017.
- [VC78] L.S. Vygotskii and M. Cole. *Mind in society : the development of higher psychological processes*. Harvard University Press, 1978.
- [VFHD16] Gabriele Valentini, Eliseo Ferrante, Heiko Hamann, and Marco Dorigo. Collective decision with 100 Kilobots : Speed versus accuracy in binary discrimination problems. *Autonomous Agents and Multi-Agent Systems*, 30(3) :553–580, 2016.
- [VYL⁺17] Jill-Jênn Vie, Florian Yger, Ryan Lahfa, Basile Clement, Kévin Cocchi, Thomas Chalumeau, and Hisashi Kashima. Using Posters to Recommend Anime and Mangas in a Cold-Start Scenario. *ArXiv e-prints*, September 2017.

- [Wat98] S. H. Watts, D. J. ; Strogatz. Collective dynamics of 'small-world' networks. *Nature*, june 1998.
- [WBK09] Ran Wolff, Kanishka Bhaduri, and Hillol Kargupta. A generic local algorithm for mining data streams in large distributed systems. *IEEE Trans. Knowl. Data Eng.*, 21(4) :465–478, 2009.
- [WCBC12] Maya Wardeh, Frans Coenen, and Trevor Bench-Capon. Multi-agent based classification using argumentation from experience. *Autonomous Agents and Multi-Agent Systems*, 25(3) :447–474, Nov 2012.
- [Web91] Noreen M. Webb. Task-related verbal interaction and mathematics learning in small groups. *Journal for Research in Mathematics Education*, 22(5) :366–389, 1991.
- [XYL15] Junyi Xu, Li Yao, and Le Li. Argumentation based joint learning : A novel ensemble learning approach. *PLOS ONE*, 10(5) :1–21, 05 2015.
- [ZKTX12] Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham, and BOWEI XI. Adversarial support vector machine learning. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12*, pages 1059–1067, New York, NY, USA, 2012. ACM.
- [Zol07] Kevin Zollman. *Network Epistemology*. PhD thesis, University of California, Irvine, 2007.
- [Zol13] Kevin J.S. Zollman. Network epistemology : Communication in epistemic communities. *Philosophy Compass*, 8(1) :15–27, 2013.

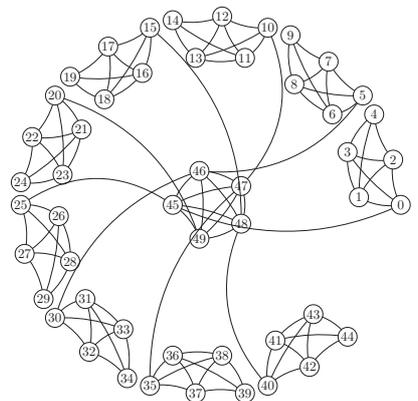
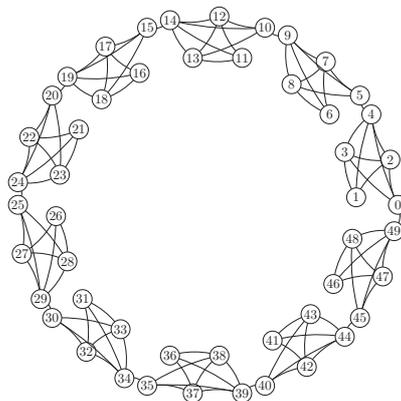
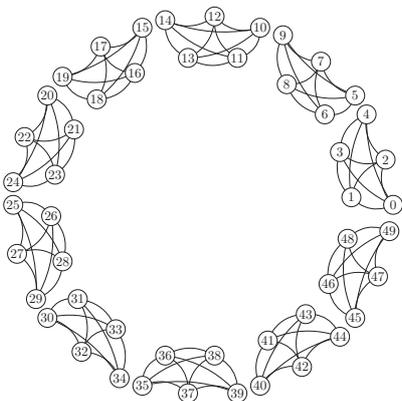
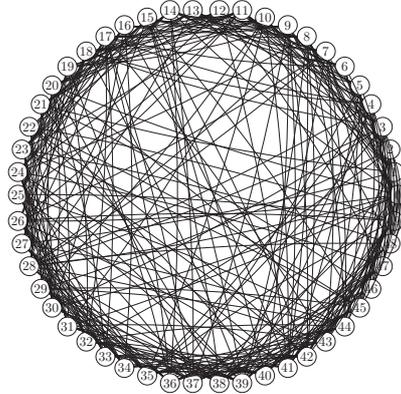
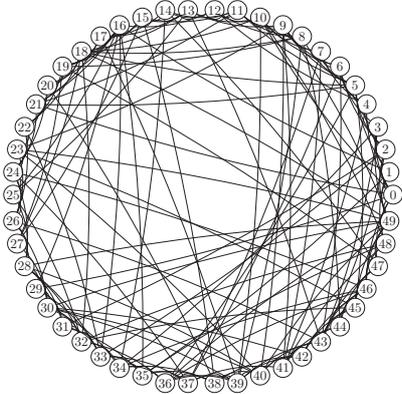
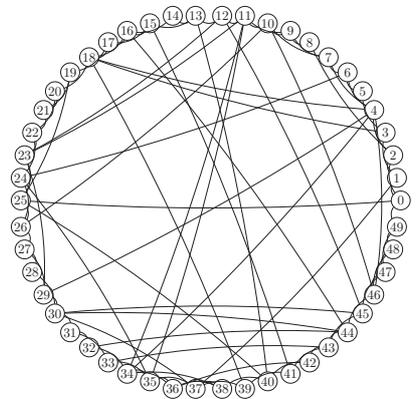
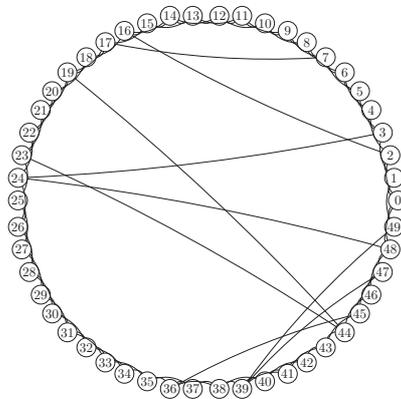
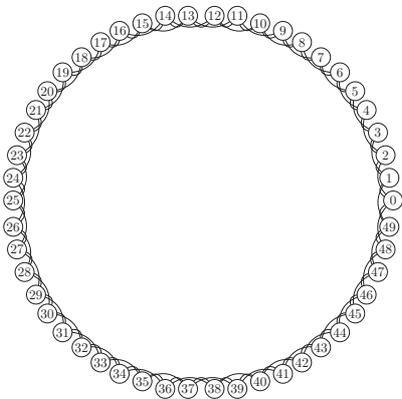
Annexe A

graphes de 50 sommets

(1ère ligne) Ligne - Roue(Multip1d008) - Étoile(Arb49)
 (2ème ligne) Arb2 - Arb3 - Arb5
 (3ème ligne) Arb7 - Bipôle(Multip2d008) - Tripôle(Multip3d008)



Graphes suite :
 (1ère ligne) Reg4 - SmW4p01 - SmW4p05
 (2ème ligne) SmW8p05 - SmW16p05
 (3ème ligne) MCluster5 - MCluster5circ - MCluster5star



Graphes suite :
 (1ère ligne) Donut (dessus) - Donut (coté)
 (2ème ligne) MultiP4d02 - MultiP5d02 - MultiP9d02
 (3ème ligne) MultiP1d02 - MultiP21d02 - MultiP21d02Sep

