



HAL
open science

Systemes de Classeurs et Communication dans les Systemes Multi-Agents

Gilles Enée

► **To cite this version:**

Gilles Enée. Systemes de Classeurs et Communication dans les Systemes Multi-Agents. Autre [cs.OH].
Université Nice Sophia Antipolis, 2003. Français. NNT: . tel-00192372

HAL Id: tel-00192372

<https://theses.hal.science/tel-00192372>

Submitted on 27 Nov 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Nice Sophia-Antipolis - UFR Sciences

École Doctorale de STIC

THÈSE

Présentée pour obtenir le titre de

Docteur en SCIENCES de l'Université de Nice Sophia-Antipolis

Spécialité : Informatique

Par

Gilles ÉNÉE

Sujet de la thèse :

Systemes de Classeurs et Communication dans les Systemes Multi-Agents

Soutenue publiquement le 6 Janvier 2003 devant le jury composé de :

Jean-Louis CAVARERO	Professeur à l'UNSA	Président
Yves DUTHEN	Professeur à l'IRIT - Toulouse I	Rapporteur
Danièle HÉRIN	Professeur au LIRMM - Montpellier II	Rapporteur
Cyril FONLUPT	Professeur au LIL - Calais	Examineur
Philippe COLLARD	Professeur à l'UNSA	Directeur de thèse
Catherine ESCAZUT	Maître de conférences à l'UNSA	Co-directeur de thèse

*Ici le temps s'arrête,
À Papa et à Luc.*

Remerciements

Les mots ne sauraient décrire la gratitude envers celles et ceux qui m'ont permis de terminer cette thèse. Je veux bien entendu parler en premier lieu de Cathy ESCAZUT et Philippe COLLARD de l'équipe Mage devenue projet TEA qui m'ont permis de développer un thème de recherche qui m'a toujours tenu à coeur : l'intelligence artificielle. Je veux également parler de Yves DUTHEN et Danièle HÉRIN qui ont accepté de relire mon mémoire dans un laps de temps très court et sans forcément bien me connaître, de Cyril FONLUPT qui a essayé de venir de loin pour me voir malgré les intempéries, et de Jean-Louis CAVARERO qui a su mener avec intelligence et humour les questions lors de la soutenance. Il y a aussi le laboratoire I3S et son équipe technique qui m'a permis d'avoir un outil de travail toujours disponible et qui a toujours accepté de m'aider malgré mes discours redondant sur la platitude des moteurs à plat.

Je tiens tout particulièrement à remercier mon collègue de bureau, Aziz, qui, bien que travaillant sur un thème différent, a toujours su communiquer avec les systèmes multi-agents hétérogènes. Je tiens également à remercier Pierre, étudiant en thèse à l'IDEFI qui m'a permis d'accomplir un travail important sur le système de classeurs que j'avais développé et qui a toujours su me guider dans mon approche du "golem" informatique. Je remercie également les philosophes qui depuis plus de 2000 ans se creusent les méninges pour découvrir les strates qui les composent.

Je voudrais remercier ma femme, Caroline, pour m'avoir toujours soutenu et écouté, même dans les moments difficiles, et surtout pour m'avoir donné notre fils, Luc. Je remercie mes chats pour m'avoir régulièrement trié des articles de recherches dans un ordre aléatoire. Je les remercie également pour leur sympathique manie de se coucher en ronronnant sur l'article que j'étais en train de lire.

Je tiens également à faire un clin d'oeil à un jeu "on-line", Everquest, qui m'a permis d'appréhender la réalité des interactions entre agents hétérogènes humains à grande échelle grâce notamment à Chenedor, Haleldar, Hypnotis, Khalbut et Cixy pour ne citer qu'eux.

Enfin je remercie tous mes "potes" motards, on/off, zebeeone, le boulet, le poireau, Pan Airlines, ZR6 2x2, lbmw pour ces quelques "arsouilles" qui m'ont permis de me changer les idées. À tous ceux que j'ai pu oublier ou que je n'ai pas nommé mais qui se reconnaissent dans cette phrase : merci.

Table des matières

1	Introduction	19
2	S’inspirer de la Nature pour Communiquer : Pourquoi ?, Comment ?	25
2.1	Les Systèmes Multi-Agents	26
2.2	Modèles de la Nature	27
2.2.1	Les étourneaux	28
2.2.2	Les fourmis	29
2.2.3	Les singes	31
2.2.4	Conclusion	32
2.3	Modèles Simulés	33
2.3.1	Éthologie synthétique	33
2.3.2	Animat	38
2.3.3	Robots	43
2.3.4	Conclusion	44
2.4	Bilan	45
3	Systèmes de Classeurs	47
3.1	Généralités	47
3.1.1	Structure	48
3.1.2	Environnement	50
3.1.3	Résolution	51
3.1.4	Évolution	51
3.2	Systèmes de Classeurs avec Algorithme Génétique	52
3.2.1	L’algorithme génétique	52
3.2.2	Systèmes de classeurs de type Michigan	55
3.2.3	Systèmes de classeurs de type Pittsburgh	58
3.2.4	XCS	62
3.3	Autres Systèmes de Classeurs	64
3.3.1	Q-Learning	65
3.3.2	Anticipatory Classifier System (ACS)	66

3.3.3	Systèmes de classeurs flous	67
3.4	Conclusion	67
4	Coordination “spontanée”	69
4.1	Définitions	69
4.2	El-Farol Problem	71
4.3	Expérimentations	71
4.3.1	Fonction de fitness	72
4.3.2	Mesures	73
4.3.3	Pittsburgh Simplifié	75
4.3.4	Paramètres	78
4.3.5	Résultats	80
4.4	Conclusion	98
5	Communiquer avec des SMA Homogènes	101
5.1	Problématique des SMA Homogènes	101
5.2	Quelques exemples	101
5.3	Distribuer la connaissance	105
5.3.1	Cloner	105
5.3.2	Application à un contrôle de carrefours routiers	106
5.3.3	Élitisme	110
5.3.4	Robustesse	114
5.4	Conclusion	117
6	Communiquer avec des SMA Hétérogènes	119
6.1	Problématique des SMA hétérogènes	119
6.2	Définition d’un modèle minimal	124
6.2.1	Structure d’un agent	125
6.2.2	Mesurer la Communication	126
6.2.3	Conclusion	128
6.3	Application	129
6.3.1	Systèmes de classeurs de type Pittsburgh simplifié	129
6.3.2	Systèmes de classeurs de type XCS	132
6.3.3	Le robot Khepera	133
6.3.4	Conclusion	136
6.4	Extension du modèle minimal	137
6.4.1	Vers la complexité	137
6.4.2	Analyse du lexique	141
6.4.3	Combinaisons de lexiques	148

TABLE DES MATIÈRES

9

6.5 Conclusion 166

7 Conclusion 169

7.1 Bilan 169

7.2 Perspectives 172

Table des figures

2.1	Zone de perceptions des animats.	39
3.1	Schéma de fonctionnement d'un système de classeurs.	48
3.2	Structure d'un système de classeurs.	49
3.3	Le Woods 101 : un environnement non Markovien.	51
3.4	Croisement mono-point.	54
3.5	Croisement n-points.	54
3.6	Mutation.	55
3.7	Mécanisme de résolution de la Bucket Brigade.	58
3.8	Exemple d'évaluation d'un système de classeurs de type Pittsburgh.	62
3.9	Illustration du XCS.	63
4.1	Comportement typique d'un SMA autonome en fonction de ses agents.	70
4.2	Fonction de fitness.	72
4.3	Résolution du problème du bar, premières générations.	80
4.4	Résolution du problème du bar, dernières générations.	81
4.5	Transition de 32 agents à 48.	81
4.6	Transition de 48 agents à 64.	82
4.7	Transition de 64 agents à 80.	82
4.8	Transition de 80 agents à 32.	83
4.9	Exemples de comportement des agents.	84
4.10	Influence du nombre de changements d'équilibre.	85
4.11	Changements d'équilibre : Agents allant au bar.	86
4.12	Changements d'équilibre : Agents restant chez eux.	87
4.13	Changements d'équilibre : Agents indécis.	88
4.14	Changements d'équilibre : Mémorisation.	89
4.15	Changements d'équilibre : Part non-déterministe.	89
4.16	Variation du nombre d'individus : Agents allant au bar.	90
4.17	Variation du nombre d'individus : Agents restant chez eux.	91
4.18	Variation du nombre d'individus : Agents indécis.	92

4.19	Variation du nombre d'individus : Mémorisation.	93
4.20	Variation du nombre d'individus : Part non-déterministe.	93
4.21	Variation du nombre de règles : Agents allant au bar.	94
4.22	Variation du nombre de règles : Agents restant chez eux.	95
4.23	Variation du nombre de règles : Agents indécis.	96
4.24	Variation du nombre de règles : Mémorisation	96
4.25	Variation du nombre de règles : Part non-déterministe.	97
5.1	Grille 3×3 de carrefours.	107
5.2	Un individu.	107
5.3	Fonction de Fitness.	108
5.4	Élitisme.	111
5.5	Élitisme Distribué.	113
5.6	Expérimentation combinée.	115
5.7	Flux croissants.	116
5.8	Flux décroissants.	117
5.9	Flux aléatoires.	118
6.1	Structure de l'agent A_1	125
6.2	Exemple de communication réussie.	126
6.3	Résultats Modèle Minimal.	130
6.4	Modèle minimal appliqué au Pittsburgh CS et au XCS.	133
6.5	Le robot Khepera	134
6.6	Exemple de monde simulé des Khépéras.	135
6.7	Résultats Khépéras et modèle minimal.	136
6.8	Résultats avec 4 mots et 4 environnements locaux.	138
6.9	Résultats avec 8 mots et 8 environnements locaux.	139
6.10	Résultats avec 16 mots et 16 environnements locaux.	140
6.11	Résultats avec 32 mots et 32 environnements locaux.	141
6.12	Homonymie et synonymie avec 4 mots et 4 sens possibles.	144
6.13	Homonymie et synonymie avec 8 mots et 8 sens possibles.	145
6.14	Homonymie et synonymie avec 16 mots et 16 sens possibles.	146
6.15	Homonymie et synonymie avec 32 mots et 32 sens possibles.	148
6.16	Succès de communication pour le scénario "Fusion" avec 4 mots.	150
6.17	Succès de communication pour le scénario "Ambassadeur" avec 4 mots.	151
6.18	Succès de communication pour le scénario "Diplomate" avec 4 mots.	151
6.19	Homonymie chez l'ambassadeur et le diplomate utilisant 4 mots.	152
6.20	Succès de communication pour le scénario "Fusion" avec 8 mots.	153
6.21	Succès de communication pour le scénario "Ambassadeur" avec 8 mots.	153

6.22	Succès de communication pour le scénario “Diplomate” avec 8 mots.	154
6.23	Synonymie dans le scénario “Fusion” en utilisant 8 mots.	154
6.24	Synonymie dans le scénario “Ambassadeur” en utilisant 8 mots.	155
6.25	Synonymie dans le scénario “Diplomate” en utilisant 8 mots.	155
6.26	Homonymie dans le scénario “Fusion” en utilisant 8 mots.	156
6.27	Homonymie dans le scénario “Ambassadeur” en utilisant 8 mots.	156
6.28	Homonymie dans le scénario “Diplomate” en utilisant 8 mots.	157
6.29	Succès de communication pour les trois scénarios avec 16 mots.	157
6.30	Succès de communication pour le scénario “Ambassadeur” avec 16 mots.	158
6.31	Succès de communication pour le scénario “Diplomate” avec 16 mots.	158
6.32	Synonymie pour l’ambassadeur et le diplomate avec 16 mots.	159
6.33	Homonymie pour l’ambassadeur et le diplomate avec 16 mots.	160
6.34	Succès de communication pour les trois scénarios avec 32 mots.	161
6.35	Succès de communication pour le scénario “Ambassadeur” avec 32 mots.	161
6.36	Succès de communication pour le scénario “Diplomate” avec 32 mots.	162
6.37	Synonymie dans les trois scénarios en utilisant 32 mots.	162
6.38	Synonymie dans le scénario “Ambassadeur” en utilisant 32 mots.	163
6.39	Synonymie dans le scénario “Diplomate” en utilisant 32 mots.	163
6.40	Homonymie dans les trois scénarios en utilisant 32 mots.	164
6.41	Homonymie dans le scénario “Ambassadeur” en utilisant 32 mots.	164
6.42	Homonymie dans le scénario “Diplomate” en utilisant 32 mots.	165
6.43	Succès de communication pour la rencontre des 2 mondes avec 4 mots.	165
6.44	Succès de communication pour la rencontre des 2 mondes avec 8 mots.	166
6.45	Succès de communication pour la rencontre des 2 mondes avec 16 mots.	166
6.46	Succès de communication pour la rencontre des 2 mondes avec 32 mots.	167
7.1	Modèle conscient.	173

Liste des tableaux

2.1	Résumé sur les Systèmes Multi-Agents dans la Nature.	32
2.2	Une matrice idéale.	35
2.3	Une matrice uniforme.	35
2.4	Table 1 de l'expérimentation de [MacLennan et Burghardt, 1994].	36
2.5	Table 2 de l'expérimentation de [MacLennan et Burghardt, 1994].	37
2.6	Table 3 de l'expérimentation de [MacLennan et Burghardt, 1994].	37
2.7	Table 1 de l'expérimentation de [Werner et Dyer, 1991], <i>Temps</i> = 100.	39
2.8	Table 2 de l'expérimentation de [Werner et Dyer, 1991], <i>Temps</i> = 5000.	39
2.9	Table 3 de l'expérimentation de [Werner et Dyer, 1991], <i>Temps</i> = 7500.	40
2.10	Table 4 de l'expérimentation de [Werner et Dyer, 1991], <i>Temps</i> = 15000.	40
2.11	Performance Prédateurs-Proie.	42
2.12	Résumé sur les Systèmes Multi-Agents Simulés.	45
3.1	Roue de la Fortune.	53
3.2	Exécution de l'algorithme de la Bucket Brigade.	59
3.3	Messagerie interne.	61
4.1	Exemple du Pittsburgh.	76
4.3	Paramètres de l'algorithme génétique.	79
4.5	Paramètre du système multi-agents.	79
5.1	Résultats de [Hogg et Huberman, 1993].	102
5.3	Paramètres de l'algorithme génétique.	109
5.5	Paramètres du système multi-agents.	109
5.6	Flux des voitures sur les routes externes.	109
6.1	Une fausse matrice idéale.	127
6.2	Comparaison des mesures.	128
6.4	Paramètres du système multi-agents.	129
6.6	Paramètres de l'algorithme génétique.	129
6.7	Exemple de deux individus en fin d'expérimentation.	131

6.8	Communications possibles.	131
6.9	Comparaison des paramètres.	132
6.10	Paramètres XCS.	132
6.11	Robots en cours d'évaluation.	134
6.12	Paramètres Khépéra.	135
6.13	Paramètres du système multi-agents.	138
6.14	Paramètres de l'algorithme génétique.	138
6.16	Paramètres du système multi-agents.	149
6.18	Paramètres de l'algorithme génétique.	150

Liste des Algorithmes

1	de la bucket brigade.	57
2	du Pittsburgh simplifié.	78
3	Élitisme Distribué	112

Chapitre 1

Introduction

Parmi toutes les possibilités qui lui étaient offertes, l'évolution a choisi de créer le cerveau humain. Elle a tout d'abord créé les "dinosauriens" qui n'ont pas réussi à "optimiser" leur environnement autrement que par le gigantisme et dont le cerveau primitif ne permettait pas d'évolution rapide. D'un coup de "baguette magique", la nature a tout effacé et recommencé avec des êtres plus "adaptés", cela se passait il y a 65 millions d'années. L'évolution se trouvant trop lente à améliorer la vie, elle a décidé que le cerveau évoluerait pendant sa vie. Ce merveilleux outil a fini par dépasser sa nature "réactive" originelle pour atteindre la "conscience". Depuis lors, il n'a de cesse de résoudre les problèmes qu'il rencontre afin d'essayer d'améliorer sa condition humaine. Cette accélération soudaine de l'évolution, qui a permis à un singe de voyager dans l'espace en moins de 100 000 années alors qu'il a fallu 2 milliards de ces mêmes années pour créer un micro-organisme, semblait avoir atteint un pallier difficilement franchissable. Pourtant l'accélération exponentielle que subit l'évolution continue. Des cerveaux (Neuman, Turing) ont créé un outil capable d'effectuer pour eux certaines tâches automatiques fastidieuses. Dès le début de cet âge, en 1956 et à Dartmouth pour être plus précis, des penseurs (cf. [Crevier, 1993]) se sont réunis afin de discuter sur le thème :

*Every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.*¹

De leurs discussions, et plus particulièrement de John McCarthy, est né le terme d'*intelligence artificielle*. Malgré toute cette belle énergie dépensée à évoluer encore et encore plus vite, l'outil de l'évolution, l'ordinateur, est encore aujourd'hui incapable de raisonner [Abelson et al., 2000] tel que nous le faisons sur des problèmes complexes. L'ordinateur peut-il, va-t-il, doit-il nous remplacer ? [Colburn, 2000]

¹Chaque aspect de l'apprentissage ainsi que tous les aspects de l'intelligence peuvent théoriquement être décrit si précisément qu'une machine puisse être créée afin de les simuler.

Philosophie

Nous allons nous placer dans un premier temps dans un cadre philosophique qui va nous permettre de mieux comprendre l'objet de notre étude. [Platon] (427-347 av. J.C.) nous donne les bases du raisonnement de manière indécidable à travers ce discours entre Socrate et Alcibiade qui illustre parfaitement son point de vue :

Soc : L'utilisateur et l'objet qu'il utilise sont différents, n'est-ce pas ?

Alc : C'est à dire ?

Soc : Par exemple, je suppose que le cordonnier utilise un outil rond, un carré, ainsi que d'autres, quand il découpe.

Alc : Oui.

Soc : Et l'outil de découpage et l'utilisateur sont vraiment différents de ce qu'il utilise pour découper ?

Alc : Bien sûr.

Soc : De la même manière ce que le fabricant de harpes utilise pour créer des harpes sera différent du fabricant de harpes lui-même ?

Alc : Oui.

Soc : Et bien c'est ce que je demandai - si l'utilisateur et ce qu'il utilise sont toujours, à ton avis, deux choses différentes.

Alc : Elle le sont . . .

Soc : Et l'Homme utilise intégralement son corps également ?

Alc : Sans aucun doute.

Soc : Et nous disons que l'utilisateur et ce qu'il utilise sont différents ?

Alc : Oui.

Soc : Donc l'Homme est différent de son propre corps ?

Alc : Il semblerait.

Soc : Mais alors, qu'est-ce que l'Homme ?

Alc : Je ne peux le dire.

Soc : Oh, mais tu peux dire - qu'il est l'utilisateur de son corps.

Alc : Oui.

Soc : Et que l'utilisateur de celui-ci doit être l'âme ?

Alc : Cela se doit.

Soc : Et le dirigeant ?

Alc : Oui.

Le corps et l'âme seraient séparés. Mais surtout que remarque-t-on dans cette discussion ? C'est qu'il y a deux niveaux dans la conscience de soi. Un qui permet de constater par des faits, qui ne sont autres que les sens, que le corps n'existe qu'à travers l'esprit. Et un deuxième niveau de conscience qui permet lui de réfléchir sur cet état de fait et par ce biais, de s'abstraire du corps matériel en

l’imaginant comme un objet. Allons plus avant dans cette distinction qui va tenir en haleine plus de deux millénaires de philosophie. Descartes [Descartes, 1637] pense également que le corps et l’esprit sont séparés :

Bien qu’il soit certain que je possède un corps avec lequel je suis intimement lié ; néanmoins, parce que, d’une part, j’ai une idée claire et distincte de moi-même, puisque je suis au demeurant seulement une entité pensante et limitée, et que, d’autre part, je possède une conscience distincte du corps, puisque c’est seulement une entité non pensante et prolongeante, il est certain que je (c’est à dire, mon esprit par lequel je suis ce que je suis) suis entièrement et vraiment distinct de mon corps, et qu’il peut exister sans lui.

Ce discours de Descartes jette les bases de la conscience et étend l’idée retranscrite par Platon. Ainsi “mon esprit peut survivre sans mon corps” nous lance-t-il. Aujourd’hui encore, nous utilisons les écrits de Descartes, son esprit a survécu à son corps même si ce n’est pas ce qu’il entendait. Donc selon lui, la conscience, et par suite le raisonnement, seraient distincts du corps. Derrière ce discours apparaît progressivement l’erreur qui a été faite dans le choix des mots, dans le choix de la communication. Le constat fait par Socrate puis Descartes n’est pas vraiment que le corps et l’esprit soient distincts. Ils ont plutôt découvert que l’esprit a la capacité de concevoir le monde qui l’entoure. Le corps n’est donc pas un simple objet réactif doté d’un esprit capable d’apprendre comme le font la plupart des animaux, car l’esprit est également capable de concevoir et d’appréhender le monde à travers lequel il existe. Dans cette hypothèse, comment est-on passé d’un système purement réactif à une entité consciente de ce qui l’entoure ? [Hume, 1750] nous donne une piste :

I shall venture to affirm, as a general proposition which admits of no exception, that the knowledge of this relation [cause and effect] is not, in any instance, attained by reasonings *a priori*, but arises entirely from experience, when we find that any particular objects are constantly conjoined with each other. ²

La conscience du monde apparaît donc par l’expérimentation de celui-ci. Le cerveau humain a ceci de particulier : sa capacité à modéliser ce qui l’entoure. Les différents sens permettent cette modélisation. Hume nous dit que la modélisation vient de l’expérimentation du monde pourtant il ne nous explique pas le mécanisme qui permet de passer des sens à la conscience.

C’est en étudiant la communication au sein des systèmes de classeurs tout au long de cette thèse que nous allons tenter de dire si un système réactif simple est capable de raisonner : c’est l’hypothèse fonctionnaliste. Mais qu’est-ce que communiquer ? Sans jeu de mots, qu’entend-on par communiquer ?

²Je peux aller jusqu’à affirmer, comme règle qui n’admet aucune exception, que la connaissance de cette relation [de cause à effet] n’est, en aucun cas, atteinte en raisonnant *a priori*, mais émerge entièrement de l’expérience, lorsque nous constatons qu’un objet est intimement lié à un autre.

Intelligence Artificielle (I.A.)

Après le point de vue philosophique, regardons la position des chercheurs en intelligence artificielle. La résolution de problèmes complexes à l'aide de l'outil informatique a permis à la communauté scientifique de se poser de nombreuses questions fondamentales. Durant une assez longue période l'ordinateur a été considéré comme apte à résoudre des problèmes que seul le cerveau humain avait été capable d'aborder jusque là. Le test de Turing qui doit permettre de reconnaître une "machine intelligente", a été vaincu par Elisa, pourtant son créateur ne reconnaît aucune intelligence à son système. Nous savons grâce à Hilbert que la plupart des problèmes complexes sont indécidables à l'aide d'une machine de Turing. Pourtant l'intelligence artificielle veut repousser les limites du modèle et étudier d'autres aspects. Le danger étant que l'ajout d'axiomes au modèle d'origine risque de rendre celui-ci incomplet au sens de Gödel. Nous risquons d'obtenir des réponses justes à partir de bases fausses : $f \Rightarrow v$. Pourtant lorsqu'on y réfléchit, seule l'introduction de la logique floue a permis de faire une avancée dans le domaine du raisonnement sans pour autant répondre à l'attente des chercheurs. L'approche centralisatrice qui consiste à utiliser un système logique pour raisonner ne semble pas la meilleure approche. C'est pourquoi, après un constat d'échec, il a fallu reconsidérer l'approche de ces problèmes difficiles à l'aide d' "Intelligences Artificielles" [Bonabeau et Theraulaz, 1994].

Aujourd'hui, on étudie, en scindant les problèmes complexes en sous-problèmes plus simples, comment on peut arriver à construire une entité qui va relier les différents sous-problèmes afin de résoudre le problème principal. Le but de cette décomposition est également de trouver, pour le chercheur en intelligence artificielle, les briques de base qui permettent à des entités simples de s'organiser afin de raisonner.

Nous voyons aisément que l'approche des philosophes et celle des chercheurs en intelligence artificielle se rejoignent sur le terrain du fonctionnalisme. L'objet de cette thèse est d'ouvrir cette porte en se mettant à la place du chercheur en I.A. et en étudiant une classe de systèmes réactifs simples : les *systèmes de classeurs*.

Guide du lecteur

Dans le second chapitre, nous justifierons nos choix pour étudier la communication dans les systèmes multi-agents. En effet, on doit se demander tout d'abord s'il est nécessaire, afin de résoudre des problèmes complexes, de se placer dans un cadre multi-agents. Ce sera l'objet d'une description de phénomènes observés sur des systèmes multi-agents que l'on trouve dans la nature. Nous verrons également comment ces phénomènes peuvent être appliqués aux environnements simulés. Dans le troisième chapitre de cette thèse, nous verrons en détail une approche qui utilise l'évolution pour la résolution de problème. Cette présentation mènera au choix de l'un de ces outils comme cadre d'étude pour la communication. On se demandera ensuite, dans le chapitre quatre, s'il est nécessaire que les agents communiquent. Ce chapitre sera l'objet de l'étude d'un problème économique et social de référence : Le problème du bar "El Farol" de Santa Fe. Et enfin s'il s'avère nécessaire d'échanger des informations entre les agents, nous étudierons les différentes méthodes possibles avec l'outil choisi.

Cette dernière étude s'étalera sur les chapitres cinq et six. Nous tirerons, en conclusion, les enseignements de l'ensemble des comportements relevés et proposerons un modèle de conscience pour les systèmes apprenants qui tente de concilier les approches des philosophes, biologistes, économistes et chercheurs en intelligence artificielle.

Chapitre 2

S'inspirer de la Nature pour Communiquer : Pourquoi ?, Comment ?

Le propre de l'homme a toujours été d'essayer d'expliquer des phénomènes naturels qu'il ne comprenait pas. C'est en observant la nature ou en faisant des expérimentations qu'il a pu se rassurer quant au monde qui l'entoure en lui donnant des propriétés appréhendables et un modèle propre à chaque particularité de celui-ci. C'est dans cet esprit que nous allons voir comment la nature peut nous inspirer dans l'étude d'un problème qui est la base de toute évolution sociale : la communication. [Labrou, Finin et Peng, 1999] remarquent simplement :

... This is not to say that one cannot have a private language, whose symbols are understood only by oneself ; however, there is not much you can do with such a language. If this private language becomes public, well, now we're talking. This leads us to a fundamental characteristic of any language : languages exist to serve a purpose, namely the communication between willing—and occasionally unwilling—participants.¹

Communiquer implique qu'il y ait au moins deux entités en relation. Ces entités qui communiquent, forment un *système multi-agents* (SMA). Nous étudierons dans un premier temps les différents systèmes multi-agents. Ensuite nous regarderons ce que peut imaginer la nature en matière de communication afin de comprendre pourquoi il faut communiquer. Enfin nous étudierons comment nous pouvons appliquer nos observations à un modèle simulé. Nous en déduirons les différents modèles exploitables pour l'expérimentation informatique.

¹Nous ne voulons pas dire qu'une personne ne peut utiliser un langage privé dont elle est la seule à comprendre les symboles ; cependant, on ne peut pas faire grand chose avec un tel langage. Si ce langage privé devient public, et bien, maintenant nous parlons. Nous en avons donc déduit une caractéristique fondamentale de tout langage : les langages existent pour servir un dessein, à savoir la communication entre des participants volontaires ou parfois involontaires.

2.1 Les Systèmes Multi-Agents

L'étude d'un problème complexe amène à la décomposition de celui-ci en sous-problèmes plus faciles à appréhender. Cette décomposition en un ensemble de sous-problèmes se nomme système multi-agents. De cette décomposition, parfois arbitraire, apparaissent des difficultés. Comment pouvons-nous être sûr que la résolution de ces sous-problèmes répond bien au problème complexe posé au départ ? Nous avons donc choisi de prendre l'image d'une pyramide à l'envers en partant de l'agent simple et en étudiant comment celui-ci pouvait évoluer pour résoudre des problèmes qui dépassent ses simples besoins. Notre démarche a consisté à étudier un type d'agent particulier et de voir comment nous pouvions l'aider à résoudre des problèmes plus difficiles.

Définition d'agent

Comme nous le rappelle le livre de [Ferber, 1995], l'agent ne peut exister qu'au travers d'un environnement ou d'un autre agent avec lequel il interagit et le reconnaît comme tel. Cet état de fait permet de parler d'identité de l'agent. Elle permet de dégager plusieurs caractéristiques propres à l'individu :

- L'agent a des buts, des intentions et des croyances qui lui sont propres.
- Ses buts, intentions et croyances sont en partie modélisés par l'apprentissage et le raisonnement ainsi que les compétences de l'agent.
- Ses compétences sont reconnues par les autres agents.
- Sa capacité d'apprendre et de raisonner est directement liée à son interaction avec l'environnement mais aussi à sa capacité à interagir avec les autres.

L'interaction avec l'environnement se fait à travers des effecteurs et des capteurs. Le but d'une population d'agents est de s'auto-organiser afin d'atteindre des buts collectifs. L'interaction de l'agent avec la collectivité l'oblige au niveau social à savoir :

- communiquer avec les autres afin d'augmenter par l'apprentissage sa vision du monde qui l'entoure et de mieux reconnaître l'individualité des autres agents.
- définir son rôle au sein de la communauté et de prévoir, anticiper l'avenir.

Les croyances de l'agent émergent de son apprentissage et finissent par devenir des certitudes. Sa capacité à raisonner l'aide également à mieux définir son rôle, ses buts et ses intentions. Le mécanisme d'intention sert à accomplir un but ou à effectuer une action dans l'immédiat ou le futur. L'intention d'accomplir un but permet de définir le rôle de l'agent. Par conséquent, ce rôle, décidé par raisonnement, permet de comprendre les intentions et progressivement le but. Il n'est donc pas toujours nécessaire d'attribuer un rôle à un agent [Sigaud et Gérard, 2000].

Toutefois en fonction de son individualité, l'agent pourra décider d'être **autonome** et d'interagir avec l'environnement afin d'apprendre et de survivre.

Dans le cas contraire, il a décidé de communiquer avec les autres. L'agent qui communique a pour but d'étendre la perception qu'il a du monde qui l'entoure ou encore de partager ses compétences

et/ou des informations.

Ce dernier point nous amène à parler de la notion de compétence. Celle-ci est propre à un système multi-agents. La compétence d'un agent est sa capacité à résoudre une tâche bien définie. Partager une compétence revient à enseigner celle-ci à des élèves. Elle implique de définir un cadre **homogène** d'échange de la compétence entre l'enseignant et ses élèves. En effet, il est impératif que la compétence soit transmise avec le moins d'erreurs possibles même si elle doit être adaptée par l'élève ensuite. Implicitement, cet échange doit amener les agents à être valorisés pour leur effort.

L'échange d'informations implique qu'une correspondance doit pouvoir être établie entre le monde et sa représentation par l'apport d'une analyse sémantique ou tout au moins d'une analyse lexicale. En effet, un agent communique une information qu'il a "capté", l'information est alors codée puis transmise à un ou plusieurs agents qui devront décoder le message. Les agents communicants peuvent alors être **hétérogènes**. L'échange d'informations permet une normalisation de la perception du monde ainsi que l'apparition d'un langage au sein de la communauté d'agents. Dans ce cas aussi, l'échange doit permettre aux deux communicants d'être revalorisés. Comme dans le cas de l'échange de compétences, l'agent peut "communiquer" une information malgré lui ou en recevoir une sans pour autant l'avoir voulu. Pourtant on ne peut pas parler de communication, il s'agit plutôt dans le premier cas d'actions sur l'environnement qui peuvent être perçues par des agents qui sont dans le second cas.

Nous pouvons déduire trois grands type d'agents :

- Les agents autonomes qui n'interagissent qu'avec l'environnement.
- Les agents homogènes qui communiquent leur savoir.
- Les agents hétérogènes qui communiquent des informations.

Quelques notions importantes concernant la communication se dégagent également :

- Il faut être plusieurs pour communiquer.
- Il faut avoir quelque chose à échanger.
- Il faut avoir l'intention de communiquer pour parler de communication.
- Implicitement, la communication doit valoriser les agents qui communiquent.

Nous supposons également qu'un agent peut ne pas se limiter à une catégorie d'agents et en changer éventuellement plusieurs fois durant sa vie. Avant d'aller plus loin dans la notion d'agent autonome / homogène / hétérogène et de communication en système multi-agents, nous allons voir comment dans la nature on peut retrouver ses différentes catégories d'agents et quelles sont leurs particularités.

2.2 Modèles de la Nature

La Nature nous permet de rapprocher modélisation et réalité sans toutefois obtenir un résultat parfait. Elle nous inspire souvent les idées qui permettent de mieux résoudre certains problèmes. A travers différents exemples, nous allons essayer de voir et de comprendre pourquoi et comment la nature a structuré les systèmes multi-agents.

2.2.1 Les étourneaux

Les étourneaux sont des oiseaux migrateurs qui traversent notre pays chaque année sous forme de nuées qui envahissent nos arbres. Ces oiseaux vivent en collectivité. Pourtant lorsqu'ils sont menacés par un prédateur, on observe un comportement typique qui ressemble à un phénomène de communication d'informations mais qui n'en est pas un. Observons-les quelque peu. Lors de leurs déplacements, les étourneaux volent en groupe de plusieurs milliers d'individus, formant ainsi un nuage imposant censé faire peur aux prédateurs. Ce comportement collectif émergent est le fruit de l'adaptation intégrée à l'évolution : c'est l'effet Baldwin [Harvey, 1997].

Lorsqu'un prédateur, tel qu'un épervier, décide d'attaquer les étourneaux, il se place au dessus d'eux et lorsqu'il s'estime en situation d'attraper une proie, il tombe sur la nuée. Les étourneaux qui sont le plus près du bord de la nuée, pris de panique s'écartent en laissant ouvert un passage dans lequel l'épervier tombe. Les étourneaux ont pour habitude de voler à une certaine distance les uns des autres. Ainsi lorsque les premiers étourneaux s'écartent, on observe un gonflement du nuage au point d'entrée de l'épervier. Ce gonflement va se propager tout au long de la traversée du nuage par l'épervier donnant ainsi l'impression que les étourneaux communiquent l'information de la position de l'épervier. Une fois le prédateur passé, le nuage reprend sa forme. D'un oeil extérieur, le nuage semble avoir ondulé dans une harmonie parfaite autour de l'épervier. En réalité, il s'agit d'un phénomène de coordination spontanée des agents. Les étourneaux proches de l'épervier paniquent et le manifestent par un cri mais ceux qui sont à l'extérieur du nuage s'écartent également alors qu'ils ne sont pas concernés.

Un comportement simple de mutualisme [Aron et Passera, 2000] qui consiste à rester en groupe à une distance de sécurité donne l'impression d'un comportement complexe de chaque agent alors que ceux-ci ne font que respecter deux règles de conduite : rester en groupe et voler dans la même direction que son voisin à une distance suffisante de celui-ci. Ces deux règles n'impliquent qu'un apprentissage limité. Ce comportement collectif évite aux agents d'avoir à développer un système plus complexe. Les agents restent autonomes même si un observateur externe pourrait penser qu'il s'agit de communication.

Pour s'en convaincre, regardons également le comportement des bancs de sardines attaqués par des pingouins. Nous observerons le même phénomène. Le prédateur est le pingouin et le banc de sardines, le système multi-agents. Lors de l'attaque, le banc de sardines ondulera autour du pingouin afin, pourrait-on penser, d'éviter celui-ci. Les sardines à l'extérieur feront la même chose sans pour autant connaître la position du prédateur en dehors d'une forme de tension dans le groupe. Au lieu d'utiliser le verbe voler, nous utiliserons le verbe nager pour les règles de conduite. Ce comportement est également un phénomène de coordination spontanée. Il semblerait donc que la coordination spontanée soit un phénomène courant dans la nature. Il fait appel à l'instinct des animaux : un comportement hérité de l'espèce. Nous pouvons en déduire un comportement typique des agents coordonnés :

- Ils sont autonomes.
- Leur but est de survivre collectivement.

- Leur compétence est de savoir évoluer au sein d'un système multi-agents à l'aide de règles simples tout en donnant l'impression d'effectuer une action concertée complexe.
- Ils n'éprouvent aucun besoin de communiquer.

2.2.2 Les fourmis

Les fourmis existent sur la Terre depuis au moins 100 millions d'années et seraient entre 120 et 130 milliards. On en connaît 10000 espèces, il en existerait en fait plus de 20000. Les fourmis ont ceci de particulier : tous les êtres d'une colonie sont apparentés et solidaires. Une colonie peut se composer de quatre individus sans reine mais peut aussi avoir plusieurs milliards d'individus avec une ou plusieurs reines. La fourmi est d'autant plus intéressante que de son comportement individuel émerge un comportement collectif. Voyons comment cet insecte social qu'est la fourmi, devient tour à tour agent autonome, homogène et hétérogène. Commençons avec les fourmis guêpe étudiées en Amazonie par [Delabie et al., 2000].

Ces fourmis ont une reine qui vit dans un tronc d'arbre en décomposition et vivent en colonie de 35 à 300 ouvrières. L'ouvrière fourrage en solitaire². Ayant une durée de vie assez courte, de l'ordre d'une quarantaine de jours, elle utilise la méthode essai/récompense pour repérer les bonnes zones fourragères. Elle va à un endroit, si il n'y a rien elle retourne travailler à la fourmilière et essaye une autre direction le jour suivant. Si elle trouve de la nourriture, elle continue à aller dans la même direction et se spécialise dans le fourrage de nourriture. Inversement si elle ne trouve toujours rien après plusieurs essais, elle finira sa vie comme ouvrière dans la colonie. Nous avons là le comportement typique d'un agent :

- qui est autonome et spécialisé dans une tâche.
- dont le but est de survivre et de faire survivre le système multi-agents auquel il appartient.
- dont la compétence est de connaître ses limites pour servir la collectivité : coopération.
- dont le besoin de communiquer est réduit au strict minimum : être reconnu par les siens.

Ce modèle général de la fourmi fourragère, permet d'expliquer un autre comportement de fourmis qui vivent en plus grande colonie et sont davantage spécialisées dans leur tâche [Pasteels et al., 1987]. Une fourmi fourragère trouve une source de nourriture et dépose, lors de son retour au nid, des phéromones³ pour indiquer le chemin vers la nourriture. Les fourmis suivantes vont être attirées par la piste et vont également marquer la piste tant qu'il y a de la nourriture [Aron et Passera, 2000]. Elles vont progressivement et indirectement échanger l'information avec les autres ouvrières fourragères de la colonie. Les termites ont le même comportement. En effet, dans l'étude de [Reinhard et Kaib, 2001], lorsqu'une termite ouvrière rencontre une source de nourriture, celle-ci, en rentrant au nid, émet une trace continue de phéromones sur le sol, indiquant le chemin jusqu'à la nourriture. L'expérimentation a également démontré que lorsqu'une termite trouve de la nourriture, elle rentre vers le nid beaucoup plus rapidement. De plus, les termites qui rencontrent par la suite cette trace, iront directement et ra-

²Pour une fourmi, fourrager consiste à aller chercher de la nourriture pour la colonie.

³Les phéromones sont des substances chimiques libérées par les glandes de la fourmi.

pidement vers la source de nourriture. À chaque aller-retour à la nourriture, les termites, comme les fourmis, marquent le chemin avec de plus en plus de phéromones.

Pour revenir au modèle multi-agents, les ouvrières fourragères constituent un système multi-agents dont tous les agents sont identiques aussi bien au niveau des buts que des moyens pour atteindre ces buts. Ces agents sont également identiques au niveau des capteurs et des effecteurs. Les fourmis se servent des phéromones comme moyen de partage de la connaissance. Elles ne communiquent pas directement à un autre agent en particulier mais elles laissent une note chimique sur un support indiquant aux autres membres de la colonie où trouver de la nourriture. Ce sont donc des agents communicants homogènes qui partagent la connaissance afin d'améliorer la performance de l'ensemble du système multi-agents. On peut assimiler le dépôt de phéromones à une relation enseignant-élève où l'enseignant écrit un livre qui indique à l'élève le chemin à suivre pour accomplir sa tâche. Les termites suivent également ce modèle d'agent :

- qui est homogène et spécialisé dans une tâche.
- dont le but est de faire survivre le système multi-agents auquel il appartient : coopération.
- dont la compétence est de savoir communiquer son savoir en plus de sa capacité à trouver des choses utiles à la collectivité et de reconnaître ce savoir.
- dont le besoin de communiquer est étendu à la transmission d'un savoir de manière indirecte : en utilisant un support.

Comparons maintenant le comportement des fourmis à celui des abeilles [von Frisch, 1967]. Lorsqu'une abeille butineuse trouve une source abondante de fleurs, elle revient à la ruche et commence une danse composée de vibrations du corps et de cercles simples ou doubles. Cette danse indique à la fois le chemin par rapport au soleil pour trouver les fleurs mais également la distance à parcourir. La première abeille qui fait cette danse est l'enseignante. La particularité de cette communication est que contrairement aux fourmis et termites, les élèves doivent être présents pour comprendre. Dès que les "élèves" abeilles qui étaient présentes ont pu constater la présence de nourriture à l'endroit indiqué par l'enseignante, elles retournent à la ruche et deviennent à leur tour professeur, ce qui permet de propager rapidement la connaissance auprès d'autres élèves. Toutes les abeilles butineuses disposent de la capacité à déchiffrer cette danse, ce sont des agents :

- qui sont homogènes et spécialisés dans une tâche.
- dont le but est de faire survivre le système multi-agents auquel ils appartiennent : coopération.
- dont la compétence est de savoir communiquer son savoir et de reconnaître ce savoir.
- dont le besoin de communiquer est étendu à la transmission d'un savoir de manière directe : visuelle et auditive.

L'échange de connaissance entre agents homogènes semble donc assez aisé car ces agents ont appris à communiquer entre eux et à normaliser et homogénéiser cette communication à l'extrême.

Revenons aux fourmis qui n'ont pas fini de nous surprendre. En effet, leur capacité à communiquer ne s'arrête pas à la catégorie sociale à laquelle elles appartiennent. Lorsqu'un événement anormal se produit, tel l'arrivée d'un intrus aux abords de la colonie, la fourmi qui a détecté l'intrusion émet

un signal d'alarme qui sera reconnu par toutes ses congénères et donnera avec précision sa position au cas où la rencontre mènerait à son décès[Aron et Passera, 2000]. Toutes les catégories de fourmis comprennent ce signal et sont capables d'en tirer les conséquences : les fourmis soldats viendront rapidement sur les lieux et les fourmis ouvrières essayeront de retourner dans le nid tout en évitant la zone d'alerte ou d'attaquer avec leur glande à poison. Ainsi, le signal d'alarme émis par des agents, hétérogènes par leur rôle dans la communauté, est compris de l'ensemble de la communauté. Pour se convaincre que les fourmis soldats et les fourmis ouvrières sont des agents hétérogènes, on peut constater que la trace de phéromones émise par les ouvrières pour indiquer le chemin à la nourriture ne poussera jamais une fourmi soldat à suivre cette piste : son rôle est différent dans la communauté. La colonie de fourmis constitue donc un ensemble d'agents :

- qui sont hétérogènes par leur physiologie et par leurs spécialités.
- dont le but est de faire survivre le système multi-agents auquel ils appartiennent : coopération.
- dont la compétence est de savoir communiquer une information importante pour la communauté et de savoir la reconnaître.
- dont le besoin de communiquer est ici lié à la survie.

2.2.3 Les singes

En Afrique, certaines colonies de singes ont développé un langage afin de permettre à chacun d'identifier son environnement sans se déplacer. Les singes verts vivent en colonie et ont appris à émettre différents signaux en fonction de la situation dans laquelle ils se trouvent afin de permettre d'identifier la nourriture mais aussi les dangers tels que les faucons, les serpents ou encore les léopards [Cheney et Seyfarth, 1982, Seyfarth, Cheney et Marler, 1980]. L'émission d'un signal par l'un des individus de la colonie permet aux autres d'adapter rapidement leurs actions à venir. La communication joue un rôle essentiel dans la survie de chacun et permet la prospérité de tous. Son coût devient nul lorsque celui qui a communiqué reçoit lui-même un signal qui va l'aider à prendre une décision pour sa survie. Ainsi les singes ont développé l'esprit social tout en gardant leur individualité. En effet, le signal émis concerne tout le monde, pas seulement un spécialiste de la recherche de nourriture ou une mère qui protège son enfant.

Le signal est ici une forme de langage commun à tous les agents :

- qui sont hétérogènes par leur rôle dans le groupe.
- dont le but est de survivre et indirectement de faire survivre le système multi-agents auquel ils appartiennent.
- dont la compétence est de savoir communiquer une information importante pour la communauté et de savoir la reconnaître. Cette information aide à la prise de décision.
- dont le besoin de communiquer est lié à la survie.

2.2.4 Conclusion

Nous avons pu observer dans la nature différents types de système multi-agents. Les étourneaux qui peuvent être considérés comme des agents autonomes lorsqu'ils réagissent à une attaque de prédateur. L'autonomie des agents indique que ceux-ci ne se préoccupent de leur voisin que parce qu'ils appartiennent au même groupe et que pour survivre, ils ont appris à rester en groupe. Il faut savoir que les étourneaux sont capables de communiquer entre eux et d'être des agents homogènes. Il est donc intéressant d'étudier le comportement d'un système multi-agents autonome afin de constater si, sans communication, le système est capable de se stabiliser autour d'un but à atteindre.

Les fourmis sont des agents également complexes. Ils peuvent être considérés tour à tour autonomes, homogènes et hétérogènes : Autonomes lorsqu'ils recherchent de la nourriture ; Homogènes lorsqu'ils indiquent le chemin de la nourriture à leurs semblables ; Hétérogènes lorsqu'ils donnent l'alerte à l'ensemble du système multi-agents auquel ils appartiennent.

Les fourmis illustrent le mieux les systèmes multi-agents et permettent de déduire un modèle minimal pour chaque catégorie de système multi-agents. La table 2.1 résume pourquoi la nature a créé différents types de systèmes multi-agents à travers les besoins et les caractéristiques de chacun d'entre eux. Les systèmes autonomes évoluent en groupe par ontogénisme devenu phylogénisme. Les systèmes homogènes permettent de diminuer visiblement le coût de l'apprentissage par l'acquisition de connaissance dans une communauté d'agents. Les systèmes hétérogènes existent parce qu'ils en retirent un bénéfice plus important que le surcoût de communication entre les agents.

	Autonomes	Homogènes	Hétérogènes
est composé de	au moins 2 agents	au moins 2 agents	au moins 2 entités (Agent / SMA)
physiquement	aucune importance	identiques	identiques / différents
rôle	survivre au sein du groupe	aider à la survie du groupe	aider à sa survie et celle du groupe
compétence	connaître ses limites	transmettre un savoir	communiquer une information
individualisme	presque complet	faible	variable
collectivement	appartient à un groupe	se voue au groupe	variable

TAB. 2.1 – Résumé sur les Systèmes Multi-Agents dans la Nature.

La fourmi nous permet aussi de constater qu'un système d'agents hétérogènes tel que la fourmière peut être éventuellement composé de plusieurs systèmes multi-agents homogènes et/ou autonomes (entités). On s'aperçoit que la notion d'hétérogénéité / homogénéité / autonomie dans la nature, dépend directement de l'observateur qui définit la problématique qui l'intéresse.

Regardons maintenant au niveau de la simulation comment sont modélisés les systèmes multi-agents. Nous nous attacherons à établir un parallèle avec la présente conclusion.

2.3 Modèles Simulés

Avant de parler de modèle de communication, nous allons définir le cadre dans lequel un modèle peut s'exprimer à l'aide de l'outil informatique. Il s'agit d'environnement simulé où les agents et leur environnement vont être "concrétisés" de différentes façons en fonction du modèle abordé. Les modèles simulés nous obligent à étudier avec soin les résultats des expérimentations avant d'en déduire leur intérêt par rapport aux systèmes multi-agents.

Nous profiterons de ce parallèle entre nature et modèles simulés afin, dans un premier temps, de nous demander de quel degré de liberté la simulation doit disposer par rapport à la réalité. Si la simulation a pour but de créer de toutes pièces des êtres simulés qui n'auront de liens avec la réalité que par les règles d'évolution qui leurs sont imposées, on parlera alors d'*éthologie synthétique*.

Si la simulation a pour but de simuler le comportement d'un être réel simplifié pour l'occasion, on parlera alors d'animal artificiel ou *animat*. Les animats nous donneront l'occasion de mieux appréhender les limites de la simulation par rapport au modèle réel.

Enfin, nous regarderons comment la simulation peut être appliquée à des modèles réels que sont les robots.

Chacune de ces catégories nous permettra de décrire des exemples où la communication est employée de diverses façons avec différents systèmes multi-agents. Enfin nous pourrons conclure sur l'intérêt des modèles simulés par rapport aux modèles naturels vu précédemment et définir une ligne de conduite de nos expérimentations sur la communication.

2.3.1 Éthologie synthétique

L'éthologie synthétique, comme son nom l'indique, a pour but d'étudier un comportement particulier propre à un animal, dans un milieu synthétique. Le but n'est toutefois pas de reproduire, par simulation, un système naturel existant. Par contre, l'étude d'un comportement implique la création d'un monde artificiel adapté à cette étude. Ce monde est composé d'*organismes synthétiques* également adaptés à l'étude. Les règles qui régissent ce monde sont simples. L'éthologie synthétique permet :

- une évolution plus rapide que dans la nature.
- d'avoir un environnement simulé beaucoup plus simple que le modèle naturel.
- d'avoir un contrôle total sur les variables qui régissent le monde artificiel.
- de mieux contrôler les expérimentations en permettant à deux populations initiales éventuellement identiques d'avoir une évolution différente.

Grâce à l'éthologie synthétique, on peut trouver les propriétés d'un système naturel par induction. Cette approche a l'avantage en outre d'éviter d'effectuer des observations sur des animaux qui ne sont plus dans leur cadre naturel et dont l'observation ne peut alors démontrer un comportement réel.

[MacLennan, 1991, MacLennan et Burghardt, 1994] s'est attaché à étudier la communication à l'aide de monde synthétique. Les organismes simulés (simorgs) dans ces expérimentations sont des

automates à états finis. Ils sont composés de règles :

$$(\Sigma, \gamma, \lambda) \Rightarrow (\Sigma', R)$$

où Σ représente le numéro d'état, γ représente la variable globale, λ représente l'état interne, Σ' est le nouveau numéro d'état et R la réponse. En résumé la règle fonctionne comme suit :

“Si je suis dans l'état Σ et que la variable d'environnement globale vaut γ et que mon état interne vaut λ alors je passe dans l'état Σ' et je fais la réponse R ”.

Les simorgs se servent d'un tableau noir comme environnement global i.e. d'une variable accessible à tous, pour prendre des décisions. Cette variable globale γ simule un médium de communication qui permet donc aux simorgs de communiquer. Durant un cycle, un simorg peut soit agir en essayant de trouver l'environnement local de celui qui a émis, soit émettre la nouvelle valeur de γ . Tous les 5 cycles, l'environnement local des simorgs est changé. Une génération représente 10 changements de leurs environnements locaux. Si l'action λ' du simorg testé correspond à l'environnement local λ de l'émetteur, il y a succès : les tests mesurent la coopération effective entre les agents. Les simorgs sont des automates complets qui disposent de toutes les configurations possibles $(\Sigma, \gamma, \lambda)$ dans leur capital de règles.

[MacLennan et Burghardt, 1994] proposent de tester leurs simorgs sans communication en changeant aléatoirement l'environnement global après chaque décision de l'un d'entre eux. Afin de mesurer les succès de communication, ils proposent d'utiliser une matrice dénotationnelle M dont le but est de représenter tous les couples (γ, λ) qui ont mené au succès. En effet, à chaque fois qu'un simorg S_R trouve l'environnement local λ du simorg S_E émetteur du message γ , une récompense est attribuée à S_R et S_E . De plus, on ajoute une unité à la matrice dénotationnelle aux coordonnées (γ, λ) pour marquer le succès. Cette matrice permet de remarquer si les simorgs utilisent des symboles distincts, i.e. un *lexique*, pour représenter leurs environnements locaux. La mesure d'*entropie* permet de vérifier si un lexique existe au sein de la population de simorgs [Shannon et Weaver, 1949]. Elle permet de distinguer le bruit dans un signal et de savoir si celui-ci est fiable. L'adaptation faite ici permet de voir s'il y a beaucoup d'erreurs, de bruit, dans les échanges entre simorgs. [Atlan, 1992] nous donne les bases de ce calcul de l'entropie H :

$$H = - \sum_{i,j} p_{ij} \times \log_2(p_{ij})$$

où p_{ij} est ainsi définie par [MacLennan et Burghardt, 1994] :

$$p_{ij} = \frac{M(i,j)}{\sum_j M(i,j)}$$

Soit M la matrice dénotationnelle, $M(i, j)$ est l'élément de la ligne i et de la colonne j . Si $\sum_j M(i, j)$ vaut 0 alors on ajoute 0 à H . Cette mesure est divisée par le logarithme népérien de 2 afin de faciliter son interprétation.

Une matrice est dite “idéale” lorsqu'elle est constituée d'un seul poids supérieur à 0 par ligne et par colonne ce qui représente alors un lexique parfait dans lequel chaque mot a un sens unique et inversement. Ce nombre est “idéalement” identique dans toute la matrice. La mesure entropique d'une matrice idéale 8×8 (cf. table 2.2) est :

		Situation $\lambda (i)$							
		0	1	2	3	4	5	6	7
S	0	0	1	0	0	0	0	0	0
y	1	1	0	0	0	0	0	0	1
m	2	2	0	1	0	0	0	0	0
b	3	3	0	0	0	0	0	1	0
o	4	4	0	0	1	0	0	0	0
l	5	5	0	0	0	0	1	0	0
e	6	6	0	0	0	1	0	0	0
$\gamma(j)$	7	7	0	0	0	0	0	0	1

TAB. 2.2 – Une matrice idéale.

		Situation $\lambda (i)$							
		0	1	2	3	4	5	6	7
S	0	0	1	1	1	1	1	1	1
y	1	1	1	1	1	1	1	1	1
m	2	2	1	1	1	1	1	1	1
b	3	3	1	1	1	1	1	1	1
o	4	4	1	1	1	1	1	1	1
l	5	5	1	1	1	1	1	1	1
e	6	6	1	1	1	1	1	1	1
$\gamma(j)$	7	7	1	1	1	1	1	1	1

TAB. 2.3 – Une matrice uniforme.

$$H_{ideale} = -8 \times \frac{1}{8} \log_2\left(\frac{1}{8}\right) = 3 \times \log_2(2) - 0 = 3.0$$

Une matrice uniforme (cf. table 2.3) est une matrice où le bruit est à son maximum dans l'établissement d'un lexique. Nous appellerons $H_{uniforme}$ la mesure d'entropie de cette matrice uniforme. On obtient alors pour une matrice 8×8 :

$$H_{uniforme} = -64 \times \frac{1}{64} \log_2\left(\frac{1}{64}\right) = 6 \times \log_2(2) = 6.0$$

De manière générale, pour une matrice de taille $n \times n$ on a :

$$H_{uniforme} = -n^2 \times \frac{1}{n^2} \log_2\left(\frac{1}{n^2}\right) = 2 \times \log_2(n)$$

$$H_{ideale} = -n \times \frac{1}{n} \log_2\left(\frac{1}{n}\right) = \log_2(n)$$

Si n s'écrit 2^m on obtient par simplification :

$$H_{uniforme} = 2 \times m$$

$$H_{ideale} = m$$

Les matrices dénotationnelles sont mises à jour lors des 50 dernières générations des expérimentations de [MacLennan et Burghardt, 1994]. La table 2.4 représente le résultat de l'expérimentation sans communication avec 8 différents symboles et 8 différents environnements locaux. La mesure d'entropie H associée est :

$$H = 5,65$$

Cette mesure semble cohérente avec l'aspect de la matrice et dénote bien une absence de lexique clair lorsqu'il n'y a pas communication. En effet, la distance⁴ de H mesurée à $H_{uniforme}$ est faible.

		Situation λ							
		0	1	2	3	4	5	6	7
S y m b o l e γ	0	180	201	27	712	149	296	254	292
	1	202	191	21	707	140	268	240	338
	2	196	199	24	699	145	284	235	290
	3	168	154	20	713	135	312	214	314
	4	200	182	15	643	149	310	226	284
	5	206	183	28	684	142	283	243	280
	6	204	191	21	676	145	290	221	310
	7	198	186	19	689	128	276	236	297

TAB. 2.4 – Table 1 de l'expérimentation de [MacLennan et Burghardt, 1994].

Lorsqu'il n'y a pas communication, le système multi-agents peut être considéré comme formé d'agents :

- qui sont pseudo-autonomes.
- dont le but est de survivre collectivement en tentant de coopérer.
- dont la compétence est inexistante en dehors de la volonté de coopérer.
- dont le besoin de communiquer est lié à la survie.

On constate d'ores et déjà que les agents autonomes provenant de l'éthologie synthétique peuvent être éloignés de leurs homologues naturels.

La table 2.5 présente les résultats avec communication. La mesure d'entropie est alors :

$$H = 3,95$$

Dans ce cas, l'entropie indique qu'il y a une forme de lexique qui émerge de la matrice. La distance de H mesurée à H_{ideale} est de seulement de 0,95. La matrice dénote toutefois quelques confusions relativement importantes qui sont apparentées à des synonymies⁵ et des homonymies⁶.

On peut décrire les agents comme étant :

- homogènes.
- dont le but est de survivre et de faire survivre la collectivité en coopérant.
- dont la compétence est d'apprendre à communiquer.
- dont le besoin de communiquer est lié à la survie.

Là encore, la liberté donnée au SMA permet de garder une certaine distance avec les modèles naturels.

La table 2.6 montre les résultats des expérimentations lorsque communication et apprentissage sont autorisés. L'apprentissage permet à un simorg qui s'est trompé d'avoir la règle responsable de

⁴La distance mesure la différence entre la valeur d'entropie mesurée pendant l'expérimentation et la valeur d'entropie maximale.

⁵Plusieurs symboles γ correspondent à la même situation λ .

⁶Un symbole γ correspond à plusieurs situations λ .

		Situation λ							
		0	1	2	3	4	5	6	7
S y m b o l e γ	0	0	0	2825	0	500	20	0	0
	1	206	0	0	505	999	231	2	0
	2	1	0	0	277	39	4935	1	2394
	3	385	1	1	94	0	0	1483	1
	4	0	292	0	0	19	555	0	0
	5	0	0	1291	0	0	144	0	0
	6	494	279	0	403	0	1133	2222	0
	7	140	2659	0	202	962	0	0	0

TAB. 2.5 – Table 2 de l'expérimentation de [MacLennan et Burghardt, 1994].

son erreur, transformée en règle coopérante. Soit λ l'environnement local à deviner et γ le symbole qui se trouve dans l'environnement global. Voici comment la "mauvaise" règle est transformée :

$$(\Sigma, \gamma, \lambda'') \Rightarrow (\Sigma', \text{Agit}(\lambda')) \text{ où } \lambda \neq \lambda' \text{ qui devient alors } (\Sigma, \gamma, \lambda'') \Rightarrow (\Sigma', \text{Agit}(\lambda)).$$

Cette forme d'apprentissage directe permet d'obtenir une mesure d'entropie H de :

$$H = 3,47$$

La table devrait donc décrire une matrice assez peu bruitée au niveau lexical. Lorsqu'on examine de près la matrice, on constate qu'il y a plusieurs poids forts dans celle-ci sans pour autant se débarrasser d'importantes homonymies et synonymies. Le gain en entropie n'est donc pas significatif. Toutefois on peut constater que le nombre total de succès de communication est très élevé. L'apprentissage permettrait essentiellement de communiquer plus fréquemment sans grandement améliorer la qualité de la communication.

		Situation λ							
		0	1	2	3	4	5	6	7
S y m b o l e γ	0	3908	29172	1287	12281	2719	1132	93	3836
	1	191	634	107	1039	0	0	2078	0
	2	4675	1306	0	37960	85	410	7306	26611
	3	0	410	0	0	0	126	1306	304
	4	0	0	353	62	575	1268	420	519
	5	36	0	46	469	0	0	0	26
	6	1075	156	0	0	0	951	0	1086
	7	0	73	54	0	2764	135	461	102

TAB. 2.6 – Table 3 de l'expérimentation de [MacLennan et Burghardt, 1994].

Les agents de ce système multi-agents sont identiques à ceux ne disposant que de la communication, à l'exception près qu'ils savent apprendre à communiquer.

Le but de ces expérimentations était d'étudier la communication au sein d'un système multi-agents. Nous avons pu voir que le degré important de liberté qu'offre l'éthologie synthétique permet de fabriquer des agents extrêmement simples afin d'étudier un comportement observé dans la nature.

Toutefois on pourrait penser que ce degré de liberté empêche de généraliser les observations. En réalité, comme [Noble, 1998] le constate, la façon dont les expérimentations sont effectuées en éthologie synthétique, est discutable :

- La façon dont la non communication est implémentée est discutable puisque non communiquer consiste à simplement modifier aléatoirement la variable globale γ à chaque cycle.
- L'imposition d'un ordre au sein des simorgs, lorsque la communication est possible, biaise les observations.
- La mesure d'entropie ne semble pas parfaitement adaptée aux observations.
- L'apprentissage est exogène.

Pourtant l'idée semble parfaitement adaptée à l'étude de la communication puisqu'elle se libère des contraintes d'un modèle observable dans la nature. Seuls les points présentés doivent être repensés. Nous allons maintenant voir une autre méthode de simulation : les animats.

2.3.2 Animat

Les animats [Wilson, 1987] sont des animaux artificiels qui permettent d'étudier des phénomènes plus complexes. Les animats évoluent dans un environnement simulé qui ressemble à un modèle naturel simplifié. Cela permet de changer les contraintes qui existent dans le milieu naturel. La communication est l'un des phénomènes les plus étudiés grâce aux animats.

L'une des expérimentations les plus enrichissantes sur la communication à l'aide d'animats a été faite par [Werner et Dyer, 1991] qui ont étudié le comportement de grenouilles artificielles.

Werner et Dyer ont créés des animats dont le besoin de communiquer est vital afin de se reproduire. Ainsi la grenouille artificielle mâle est aveugle mais sait se déplacer. La femelle grenouille voit mais ne peut se déplacer. Grâce à un cri adapté, elle doit permettre au mâle qu'elle voit de venir à elle. Le mâle quant à lui, doit apprendre à interpréter les cris entendus afin d'arriver à la femelle pour que les deux se reproduisent.

Pour simplifier l'expérimentation les animats se trouvent sur une grille de 200×200 cases. 800 mâles et 800 femelles sont placés aléatoirement sur la grille occupant ainsi 4% de la grille. C'est un réseau de neurones qui représente les agents. Les femelles sont capables d'identifier la direction et l'endroit où se trouve le mâle le plus proche mais ne peuvent pas se déplacer. Les mâles peuvent entendre les différents signaux émis par la femelle la plus proche. Ils sont capables d'avancer, rester sur place ou encore de tourner à gauche ou à droite. La figure 2.1 montre les limites de la zone de perception des animats. Lorsqu'un couple se forme, i.e. le mâle et la femelle sont sur la même case, il se reproduit en suivant le schéma de reproduction des algorithmes génétiques. De leur union naît un mâle et une femelle qui vont remplacer un couple moins performant. Les parents sont ensuite replacés aléatoirement dans la grille pour leur permettre de tester d'autres configurations.

Les tables 2.7, 2.8, 2.9 et 2.10 mettent en lumière l'évolution de l'interprétation du signal des femelles par les mâles au cours d'une expérimentation. Le signal a été ici transformé en valeur entière. Pour chaque signal est indiqué le pourcentage de mâle qui décide de faire l'une des actions possibles :

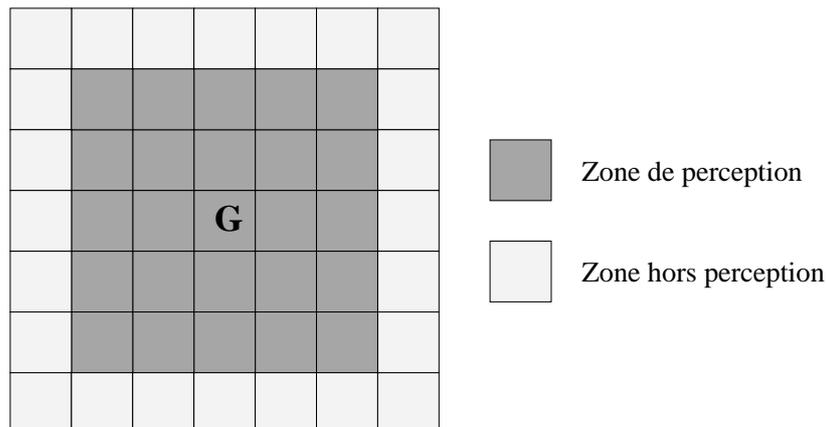


FIG. 2.1 – Zone de perceptions des animats.

avancer, tourner à droite ou à gauche ou ne rien faire. Un pas de temps correspond à l'action de toutes les femelles puis de tous les mâles. C'est à dire à la lecture de l'environnement par les femelles, à l'émission d'un signal correspondant à ce qu'elles voient puis à l'écoute par les mâles des signaux émis et à un choix de mouvement à effectuer.

Signal	Avancer	Tourner à Droite	Tourner à Gauche	Ne rien faire
0	25	38	9	28
1	19	25	31	25
2	28	22	26	24
3	29	25	29	17
4	25	27	26	22
5	25	25	26	24
6	19	20	31	30
7	22	27	21	30

TAB. 2.7 – Table 1 de l'expérimentation de [Werner et Dyer, 1991], *Temps* = 100.

Signal	Avancer	Tourner à Droite	Tourner à Gauche	Ne rien faire
0	74	20	5	1
1	81	12	7	0
2	67	18	14	1
3	79	11	19	1
4	80	7	12	0
5	75	11	14	0
6	56	21	23	0
7	70	14	16	0

TAB. 2.8 – Table 2 de l'expérimentation de [Werner et Dyer, 1991], *Temps* = 5000.

Signal	Avancer	Tourner à Droite	Tourner à Gauche	Ne rien faire
0	98	2	0	0
1	99	0	0	1
2	98	1	1	0
3	100	0	0	0
4	100	0	0	0
5	99	1	0	0
6	98	1	1	0
7	99	0	1	0

TAB. 2.9 – Table 3 de l'expérimentation de [Werner et Dyer, 1991], *Temps* = 7500.

Signal	Avancer	Tourner à Droite	Tourner à Gauche	Ne rien faire
0	97	2	0	1
1	100	0	0	0
2	98	1	1	0
3	98	0	1	1
4	100	0	0	0
5	22	77	1	0
6	5	2	93	0
7	97	0	3	0

TAB. 2.10 – Table 4 de l'expérimentation de [Werner et Dyer, 1991], *Temps* = 15000.

Les signaux provenant des femelles sont dans un premier temps incompris et mènent à toutes sortes de comportements (voir table 2.7). Puis, le temps passant, les agents mâles découvrent que rester sur place ne rapporte rien (voir table 2.8). Ensuite ils constatent que d'avancer lorsqu'ils reçoivent un signal finit toujours par payer (voir table 2.9) et enfin ils se mettent à écouter les femelles et finalement deux signaux distincts leur permettent de se diriger vers les femelles (voir table 2.10). Le signal 5 signifie alors "tourner à droite" et le signal 6, "tourner à gauche".

L'expérience de Werner et Dyer montre que lorsque le cadre d'étude est bien défini, le comportement des animats s'apparente à celui que l'on pourrait imaginer avec des animaux. De plus les animats minimalisent le besoin de communication en le réduisant à deux signaux "tourner" qui deviennent communs à toute l'espèce. De cette minimalisation, une nouvelle expérimentation est mise en place afin d'étudier l'évolution d'un dialecte dans la population d'animats.

Dans cette étude de la formation des dialectes, [Werner et Dyer, 1991] autorisent 4 signaux aux femelles. Le but est de suivre le nombre de mâles qui effectuent une action donnée pour un signal donné. Les résultats sont présentés à l'aide du format suivant : $N : A_1A_2A_3A_4$. N est le nombre d'animats mâles concernés par la règle de comportement et A_i l'action qu'ils décident de choisir pour le signal i provenant des femelles. Par exemple, le résultat 7 : 3120 signifie que 7 animats mâles tournent à droite (3) avec le signal 1, avancent (1) avec le signal 2, tournent à gauche (2) avec le signal 3 et ne font rien (0)

lorsqu'ils perçoivent le signal 4. Werner et Dyer ont pu ainsi découvrir que la combinaison qui sera vainqueur ($n : 1313$) n'apparaît comme challenger qu'au bout d'un $Temps = 14000$. Cette combinaison va s'estomper avant de revenir au $Temps = 20000$ pour finir par dominer au $Temps = 40000$. La communication évolue donc avec le temps. Le comportement final n'est pas déterminable à l'avance. La communication, s'effectuant en environnement non "bruité", explique la stabilisation d'un choix de comportement commun des mâles pour chaque signal provenant des femelles. Les agents sont :

- hétérogènes par leur sexe et par leurs capacités différentes et limitées.
- dont le but est de survivre en coopérant.
- dont la compétence est d'apprendre à communiquer.
- dont le besoin de communiquer est lié à la survie.

Parallèlement, Werner et Dyer montrent que la non-communication, à savoir quand le mâle n'entend rien, fonctionne mieux dans un premier temps. Effectivement, la communication a un coût qui est en fait ici le temps nécessaire aux mâles pour comprendre les femelles. Ce temps peut être crucial pour la survie des animaux communicants car ils se trompent plus durant cette phase. Une fois ce laps de temps franchi, la communication devient rentable.

Les agents non-communicants sont :

- autonomes.
- leur but est de survivre collectivement en tentant de coopérer.
- dont la compétence est inexistante en dehors de la volonté de coopérer.
- dont le besoin de communiquer est inexistant.

Werner et Dyer ont utilisé les animaux de façon remarquable afin d'étudier l'évolution de la communication. La simplification qu'ils ont faite du modèle réel permet de bien comprendre le cadre de leur étude. Par contre il est impossible de généraliser leurs conclusions à toute la communication. Leur modèle est en effet trop dépendant du système multi-agents dans lequel les agents évoluent.

L'étude du problème Prédateurs-Proie [Edwin De Jong, 1997] fournit également un cadre d'étude intéressant des animaux. Voici un exemple dans lequel 4 prédateurs doivent se coordonner sur une grille 30×30 pour encercler une proie qui se déplace aléatoirement. L'encercllement consiste à avoir un prédateur au nord, à l'est, à l'ouest et au sud. Quatre méthodes de communication sont étudiées. La première, nommée *local control*, consiste à signaler aux autres prédateurs lorsque l'on est dans une position de capture. Une seconde, nommée *distributed control*, permet aux prédateurs de communiquer leurs intentions concernant la position de capture qu'il souhaite occuper. Une troisième, la *central control*, permet à un agent de contrôler les autres. Enfin, celle proposée par De Jong, consiste à choisir le mouvement qu'effectue le prédateur en fonction de la distance de celui-ci à la proie ainsi que son écartement à l'angle idéal par rapport aux deux autres prédateurs les plus proches. Cette fonction *Eval* permet d'idéaliser le positionnement de chaque prédateur par rapport à son voisin et donc par rapport à la proie. Elle permet également de réduire la distance à la proie :

$$\phi_{opt} = \frac{2\pi}{\text{Nombre}(\text{Prédateurs})} = \frac{\pi}{2}$$

$$E_{max} = 2\pi - \phi_{opt} = \frac{3\pi}{2}$$

$$Eval(A_p, A_q) = \delta \times Dist(A_p, A_q) + (1 - \delta) \times Rpartition(A_p, A_q)$$

$$Dist(A_p, A_q) = \varepsilon \times equidistance(A_p, A_q) + (1 - \varepsilon) \times distance(A_p, A_q)$$

$$equidistance(A_p, A_q) = 1 - \frac{|A_p.d - A_q.d|}{\max(A_p.d, A_q.d)}$$

$$distance(A_p, A_q) = 1 - \frac{A_p.d + A_q.d}{2r\sqrt{2}}$$

$$Rpartition(A_p, A_q) = 1 - \frac{|\phi_{opt} - ((A_p.\phi - A_q.\phi) \bmod 2\pi)|}{E_{max}}$$

où ϕ_{opt} est l'angle qu'il doit y avoir entre chaque prédateur et la proie pour être en situation de capture. E_{max} est l'erreur maximale dans le calcul de l'angle. $A_p.d$ est la distance de l'agent p à la proie. $A_p.\phi$ est l'angle de l'agent p à la proie. La fonction *Rpartition* permet de déterminer si deux prédateurs A_p et A_q ont un angle d'approche de la proie proche de l'idéal ϕ_{opt} . La fonction *distance* donne la distance de deux prédateurs par rapport à la proie. *equidistance* permet de normaliser la distance des deux prédateurs en paramètre par rapport à la proie. ε et δ sont des paramètres qui permettent d'ajouter un poids pour chaque fonction servant au calcul de *Eval* et *Dist*. *Dist* permet d'évaluer la distance de deux prédateurs à la proie et *Eval* si ces deux prédateurs sont en plus bien répartis pour capturer la proie.

Les résultats sont présentés dans la table 2.11 en utilisant les mesures et résultats du *MaxCoordinationPredator* de [Edwin De Jong, 1997] ainsi que ceux de [Stefens et Merx, 1990].

	Capturés %	Succès %	Succès efficacité %
Local control	10,00%	33,33%	31,9%
Distributed control	83,33%	90,00%	69,7%
Central control	100%	100%	64,10%
MaxCoordinationPredator	100%	100%	66,70%

TAB. 2.11 – Performance Prédateurs-Proie.

Ces résultats montrent clairement que c'est la qualité de l'information fournie [Tan, 1993] qui détermine la performance de la communication. Un agent qui arrive à élaborer une information complexe et pertinente a plus de chances que la communication le valorise. Cette étude est donc intéressante de ce point de vue. Elle permet de voir la réaction d'un système multi-agents lorsque l'information communiquée change. Les agents du SMA *MaxCoordinationPredator* sont :

- homogènes.
- dont le but est de survivre et de faire survivre la collectivité en coopérant.
- dont la compétence est d'apprendre à communiquer une information pertinente.
- dont le besoin de communiquer est lié à la survie.

Il faut cependant constater que l'obtention d'une information pertinente est ici faite par une fonction élaborée qui ne reflète pas les possibilités supposées restreintes des agents simulés.

Nous allons aborder maintenant le problème du dilemme du prisonnier qui est un problème connu de la théorie des jeux et qui permet également d'étudier le comportement de "super" animats qui apprennent soit à coopérer, soit à mentir à tour de rôle. Deux prisonniers sont confrontés et doivent décider s'ils coopèrent ou non avec l'autre prisonnier en fonction de stratégies pré-déterminées. Cer-

taines stratégies rendent le prisonnier autonome car il ne cherche à avoir aucune information sur les précédentes positions de son adversaire. Lorsqu'une stratégie implique d'avoir des connaissances sur l'autre prisonnier, il s'agit de mémorisation des précédentes attitudes de celui-ci : Il n'y a pas communication. Cet exemple permet donc d'étudier des agents qui sont :

- autonomes.
- dont le but est de survivre.
- dont la compétence est d'avoir une stratégie pour augmenter sa récompense.
- dont le besoin de communiquer est inexistant.

En fait l'intérêt du problème du dilemme du prisonnier est de fixer les limites de l'autonomie en fonction de paramètres externes. Ces paramètres sont les récompenses associées à l'attitude du prisonnier :

- R est la récompense lorsque les deux prisonniers coopèrent.
- P est la récompense lorsque les deux prisonniers décident de ne pas coopérer.
- T est la récompense pour ne pas avoir coopéré tandis que l'autre prisonnier aura la récompense S pour avoir coopéré.

Pour qu'il y ait dilemme sans pour autant que la tentation soit le seul choix possible, il faut que :

$$T > R > P > S$$

$$2R > T + S$$

On peut facilement inverser le problème en choisissant :

$$S > P > R > T$$

$$2P > T + S$$

L'interprétation est inversée, pourtant le problème est exactement le même.

L'expérimentation de [Beaufils et al., 1997] montre que la meilleure stratégie pour les prisonniers consiste à faire de la répression en escalade : c'est la méthode *gradual*. [Miller et Butts, 1998] ont proposé une version modifiée du problème dans laquelle les deux prisonniers peuvent communiquer. Toutefois les réglages de T , R , P et S qu'ils ont choisi, poussent les prisonniers à coopérer. Le choix d'une stratégie n'est valide que dans le cadre des paramètres de récompense précédemment définis. L'autonomie de l'agent est fortement limitée car on lui impose une stratégie dont la performance dépend de paramètres externes. On imagine qu'en environnement dynamique dans lequel les paramètres T , R , P et S changent, les performances des prisonniers seraient toutes autres.

Les animats offrent de larges perspectives d'étude de la communication en système multi-agents. Les animats n'ont pour faiblesse que le modèle auquel ils sont rattachés. Nous allons nous rapprocher un peu plus des modèles réels pour étudier les SMA à l'aide des robots.

2.3.3 Robots

Les robots sont des modèles physiques qui permettent d'étudier certains phénomènes de manière concrète.

[Kaplan, 2001] a fait une étude sur la communication et la naissance d'un langage chez les robots. L'expérience se nomme "expérience des têtes parlantes". Le principe est simple : à l'aide d'une

caméra, un robot identifie par segmentation un objet simple fixé sur un tableau blanc. Il catégorise cet objet à l'aide des différents canaux sensoriels dont il dispose (HPOS, VPOS, HEIGHT, WIDTH, AREA, RECT, YELLOW, BLUE, GREEN, RED, LIGHTNESS). Cette catégorisation n'utilise que certains de ces sens. Enfin, il émet un mot correspondant à cette catégorisation. Un autre robot reçoit le message sans bruit et doit à partir de celui-ci et de la scène, identifier l'objet choisi par le premier robot. Pour cela, il associe le mot à une catégorisation qui peut être différente de celle du premier robot. Cette catégorisation permet d'obtenir un segment du tableau blanc. Si le segment pointé est le plus proche de l'objet choisi au départ, le jeu est alors un succès. En cas d'échec, c'est à dire si aucun segment n'est identifié ou bien encore si plusieurs segments sont identifiés grâce au mot ou enfin si le segment identifié n'est pas celui le plus proche, le premier robot montre le segment qu'il fallait choisir au second robot. Celui-ci émet alors une nouvelle hypothèse sur le sens du mot et effectue alors une diminution du score de la précédente hypothèse qui était fautive. L'environnement, à savoir le tableau blanc, est changé à intervalle régulier. Les objets qui se trouvent sur le tableau sont des cercles, rectangles, carrés et triangles de différentes couleurs. Les agents sont composés de tables qui pour un mot donné associe des intervalles de valeurs de capteurs à l'aide d'un arbre de décision.

Une première phase simulée a permis de valider ce modèle de communication et d'obtenir un succès expérimental de près de 100% une fois que le lexique s'est stabilisé.

Lors du passage au monde réel et afin d'ajouter quelques difficultés à l'expérience, les robots sont situés dans des lieux géographiques différents et des humains peuvent interagir pour donner un mot correspondant à ce que le locuteur observe. L'étude a permis de montrer que le langage, plus exactement le lexique, des robots évoluait dans le temps en taille et en diversité. D'autre part, certains mots choisis par les humains n'avaient pas le sens escompté. De plus, certains mots ont changé de sens au cours de l'expérience. A la fin de l'expérimentation, le lexique formé est pertinent à plus de 80% même si les agents se trompent à plus de 40%. Par exemple le mot "rouge" signifie que le capteur RED est dans l'intervalle [0.5, 1.0].

Cette étude a montré que le monde réel entraîne des difficultés de compréhension à cause d'un bruit involontaire, à savoir la position des caméras, l'éclairage et la qualité des capteurs et des viseurs. Le passage au monde réel demande donc des agents disposant d'une forte adaptabilité et qui sont capables de gérer le bruit qu'il y a par rapport au modèle simulé. Les agents sont ici :

- homogènes.
- dont le but est de survivre en coopérant.
- dont la compétence est d'apprendre à communiquer une information pertinente et à la comprendre.
- dont le besoin de communiquer est lié à la survie.

2.3.4 Conclusion

Nous avons étudié trois différents types de modèles simulés. Des modèles totalement abstraits à travers l'éthologie synthétique permettent d'étudier un phénomène sans s'attacher à un modèle réel.

Les agents sont alors des êtres simulés créés de toutes pièces qui évoluent grâce à des règles simples liées au phénomène à étudier. Il faut cependant être prudent dans le choix des agents ainsi que dans la façon d'étudier le phénomène pour ne pas revenir à un modèle naturel qui biaiserait les résultats.

Nous avons ensuite vu les animats qui permettent d'étudier un phénomène à l'aide d'êtres artificiels provenant de modèles naturels mais qui sont simplifiés pour la simulation. Les animats semblent être un excellent intermédiaire entre le modèle provenant de l'éthologie synthétique et la réalité. Ils permettent de valider des choix et également de se poser d'importantes questions quant au bien fondé de certains de ces choix.

Enfin le modèle réel, qui est représenté par les robots, permet d'inclure le bruit, provenant de la réalité, dans le modèle simulé. Il exige que les agents disposent d'une forte adaptabilité pour qu'ils puissent continuer à exprimer les propriétés révélés lors des deux étapes précédentes.

La table 2.12 permet de constater que les différents types d'agents des systèmes multi-agents simulés créés par les chercheurs sont souvent proches des modèles naturels.

	Autonomes	Homogènes	Hétérogènes
est composé de	au moins 2 agents	au moins 2 agents	au moins 2 entités
physiquement	aucune importance	identiques	identiques / différents
rôle	survivre au sein du groupe	aider à sa survie et celle du groupe	aider à sa survie et celle du groupe
compétence	connaître ses limites / stratégie	transmettre un savoir	communiquer une information
individualisme	presque complet	variable	variable
collectivement	appartient à un groupe	variable	variable

TAB. 2.12 – Résumé sur les Systèmes Multi-Agents Simulés.

Les modèles simulés fournissent un cadre idéal pour l'étude de la communication. L'éthologie synthétique, les animats et les robots ont permis d'étudier aussi bien des systèmes multi-agents avec communication que sans communication. Nous utiliserons ces trois types de modèles simulés pour nos expérimentations.

2.4 Bilan

Nous venons de voir les différents types de système multi-agents. Nous avons également observé pourquoi la nature utilise la communication ou non dans ces différents systèmes multi-agents. Enfin nous avons étudié les différents moyens dont nous disposons pour appliquer les modèles provenant de la nature à la simulation. Ce dernier point nous a permis de définir un degré de liberté du modèle simulé par rapport à la réalité. Ce degré va d'important avec l'éthologie synthétique jusqu'à faible avec les robots.

La nature nous a fourni différents modèles que nous avons pu retrouver en partie dans les modèles simulés. Tout d'abord, le dilemme du prisonnier, les fourmis, les animats grenouilles, les étourneaux

et les sardines nous rappellent qu'il est nécessaire d'étudier l'agent lorsqu'il ne communique pas. Cette étude permet de connaître les capacités de l'agent lorsqu'il est autonome face à un problème. Nous expérimenterons dans le chapitre 4 le comportement de tels agents.

Cette première étude sur la non-communication achevée, nous regarderons dans le chapitre 5 comment ce même agent peut échanger du savoir lorsqu'il devient homogène. Ce travail a été inspiré par l'étude du comportement des fourmis.

Enfin nous verrons comment intégrer l'hétérogénéité à nos agents dans le chapitre 6. Nous verrons également comment ils arrivent à communiquer et à établir un lexique entre eux en nous plaçant dans le cadre de l'éthologie synthétique.

Pour appliquer ces différents modèles de communication, il nous faut définir un cadre pour la simulation. C'est l'objet du chapitre 3 qui va présenter un système capable d'apprendre et de réagir aux stimulations provenant de l'environnement : les systèmes de classeurs.

Chapitre 3

Systemes de Classeurs

Holland [Holland, 1992] a introduit des systemes capables d'interagir avec l'environnement dans lequel ils evoluent. Ces entites sont dotees de capacites d'apprentissage que nous allons etudier tout au long de ce chapitre, ainsi que d'autres proprietes caracteristiques. On les nomme "Systemes de Classeurs" (SC) [Holland et al., 2000]. Depuis leur introduction en 1978, les systemes de classeurs ont subi plusieurs metamorphoses. Nous allons voir que des differentes classes de systemes de classeurs qui existent, on peut degager plusieurs points communs aussi bien au niveau de leur structure que de leur fonctionnement. Les systemes de classeurs possedent une expressivite accrue par leur composition et leur relative simplicite conceptuelle. L'etude de ces points communs va nous amener a etudier leurs particularites afin de mieux apprehender leur fonctionnement.

3.1 Generalites

Les systemes de classeurs sont assimilables a des systemes experts. Ils sont essentiellement destines a resoudre des problemes reels, dynamiques et souvent non-deterministes car fortement bruites. Les systemes de classeurs necessitent une phase d'apprentissage en environnement simule avant d'etre appliques au modele reel. Ils acquierent au long de cette phase d'apprentissage simule une base de connaissance qui leur servira a prendre une decision lorsqu'ils seront utilises en temps que systemes experts. Notons que l'apprentissage ne s'arrete pas avec la simulation. Il existe de nombreux points communs entre les differents types de systemes de classeurs. Ces caracteristiques sont a la fois leur force et leur faiblesse. Regardons maintenant plus en detail le schema de fonctionnement (fig. 3.1) d'un systeme de classeurs :

1. L'environnement reflète un état qui est lu par les "capteurs" du système de classeurs.
2. Le système de classeurs constitue, à partir de la population qui le compose, un sous-ensemble de classeurs ou règles (cf. § 3.1.1) qui peuvent agir sur l'environnement i.e. classeurs qui "comprennent" le signal émis par l'environnement.
3. A partir de cet ensemble de classeurs appelé également "Match Set", il détermine une action à

entreprendre.

4. Le système de classeurs agit sur l'environnement.
5. L'environnement sanctionne positivement ou négativement le système de classeurs.
6. C'est le module d'*apprentissage / évolution* qui gère la façon d'utiliser cette punition / récompense pour déterminer la nouvelle valeur représentative du ou des classeurs ayant participé à l'action. C'est ce même module qui va utiliser cette nouvelle valeur afin de faire évoluer les classeurs.

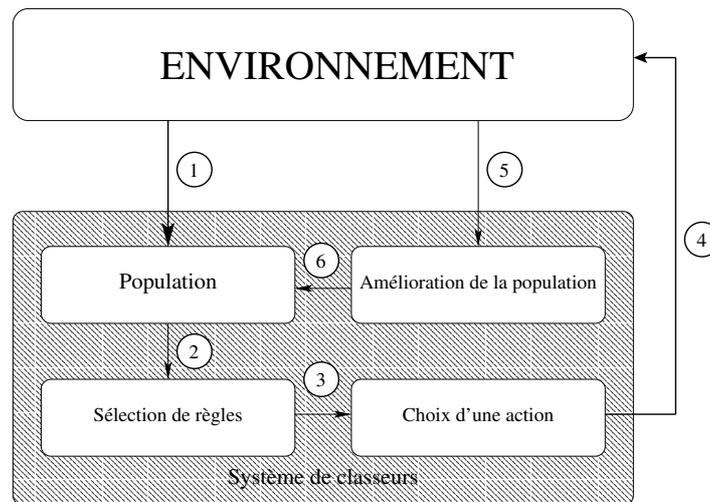


FIG. 3.1 – Schéma de fonctionnement d'un système de classeurs.

A la fin de l'apprentissage, la population contient potentiellement la(les) solution(s) au problème. Nous entrevoyons que le module d'apprentissage / évolution va permettre au système de "converger" vers la solution en répartissant la sanction de l'environnement dans le système de classeurs. D'autre part, le mécanisme de résolution est différent selon le type de système de classeurs. Ce mécanisme de résolution regroupe le sous-ensemble de classeurs sélectionnés grâce au signal de l'environnement ainsi que la détermination de l'action à entreprendre sur l'environnement. Nous allons détailler point par point le fonctionnement de l'ensemble en expliquant tout d'abord la structure commune aux différents systèmes de classeurs. Ensuite nous décrirons les environnements et leur signal. Enfin, nous poserons les bases communes des mécanismes de résolution et d'évolution des systèmes de classeurs.

3.1.1 Structure

Un système de classeurs est composé de :

- Une population d'individus.
- Chaque individu est lui-même soit un classeur, soit un ensemble de classeurs aussi appelé *structure de connaissance*.

- Un individu a une force de *fitness*, ou force d'adaptation, qui est la valeur représentative de cet individu par rapport à la solution et qui évolue en fonction des sanctions de l'environnement.
- Un classeur a une partie condition et une partie action : les classeurs sont appelés également des *règles de production*.
- Les règles de production sont des chaînes de caractères définies sur un alphabet.
- La partie condition représente les critères qui vont déterminer si le classeur peut être activé ou non à un instant t i.e. elle représente les **capteurs** du système de classeurs.
- La partie action sert à déterminer le comportement du système de classeurs i.e. elle représente les **effecteurs** du système de classeurs.

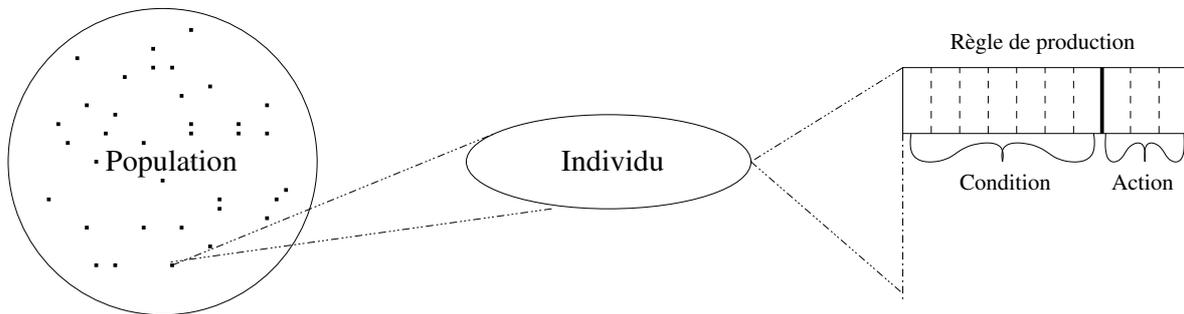


FIG. 3.2 – Structure d'un système de classeurs.

La figure 3.2 permet de mieux appréhender la structure d'un système de classeurs. Cette structure assez générale permet de regrouper l'ensemble des différents systèmes de classeurs. La valeur représentative d'un individu reflète sa performance lors de son interaction avec l'environnement. On constate que plusieurs classeurs peuvent être activés lors d'une interaction. Il y a donc un phénomène de parallélisme implicite au sein des systèmes de classeurs. Pour l'ensemble de cette présentation des systèmes de classeurs, nous supposons que :

- La chaîne de caractère composant chaque classeur est de longueur fixe : λ
- La partie condition est définie sur un alphabet ternaire : $\{0, 1, \#\}$
- Le caractère '#' signifie soit 0, soit 1.
- La partie action ainsi que le signal émis sont quant à eux définis sur un alphabet binaire : $\{0, 1\}$

La partie condition d'un classeur représente en fait la condition d'une instruction *si* et la partie action le *alors* de cette même instruction. Par conséquent, un classeur est sélectionné *si* sa partie condition *match*, i.e. correspond, parfaitement le signal de l'environnement. Première remarque, la partie condition et le signal émis ont donc la même longueur σ avec $\sigma < \lambda$. Deuxième remarque, une condition C ($C_0C_1\dots C_\sigma$) satisfait le signal S ($S_0S_1\dots S_\sigma$) si :

$$\forall \kappa \in 0, 1, \dots, \sigma (C_\kappa = S_\kappa) \vee (C_\kappa = \#)$$

Par exemple, la condition 0#1 sera satisfaite pour les signaux 001 et 011 mais pas pour le signal 000. Dernière remarque, le déclenchement de la partie action des classeurs sélectionnés dépend de la

catégorie du système de classeurs.

3.1.2 Environnement

Cette structure particulière induit une interaction avec l'environnement par les capteurs et les effecteurs du système de classeurs. L'environnement interagit avec le système de classeurs en envoyant des messages ou signaux qui correspondent à ce que les capteurs vont lire. Un environnement représente à la fois le problème à étudier et la fonction de récompense ou fonction de fitness. Il est donc le seul repère pour apprendre du système de classeurs. Il existe plusieurs catégories d'environnement :

Statique

Le terme “statique” signifie : qui n'évolue pas dans le temps. Par exemple une fonction sans variable temporelle à optimiser comme l'optimisation de la forme d'une lentille est statique. Un autre exemple peut être l'apprentissage de la/les solution(s) qui mène(nt) à la sortie d'un labyrinthe pour un animat (cf. § 2.3.2). Dans ces environnements, la tâche consiste à adhérer le plus possible à la fonction de fitness en apprenant les différentes “meilleures” actions pour chaque signal possible.

Dynamique

Optimiser la gestion des feux d'un carrefour routier constitue un environnement dynamique. En effet, le flux de voitures est un phénomène temporel difficilement prévisible. Le système de classeurs doit alors envisager des solutions afin de minimiser le temps d'attente des automobilistes au carrefour. La gestion de trafic aérien est un autre exemple d'environnement dynamique. Ces environnements imposent une adaptation forte du système de classeurs qui doit rapidement trouver une bonne solution dans un environnement souvent fortement bruité. En effet, les environnements dynamiques entraînent souvent la panne de capteurs et exigent des classeurs de “panique” efficaces. On parle plus facilement de bonne solution que de meilleure solution car la problématique en environnement dynamique est surtout d'optimiser dans le temps et pas seulement à un instant t .

Markovien / non Markovien

Un environnement est Markovien lorsque pour chaque message donné de l'environnement, il n'y a qu'une action qui mène au succès. Inversement, on parle d'environnement non Markovien (cf. figure 3.3) lorsqu'il existe des situations identiques en apparence mais dont l'action qui mène au succès est différente : non-déterministe. Un environnement non Markovien peut être statique ou dynamique. Le problème qu'engendre le non déterminisme est que le système va mettre du temps à oublier la mauvaise réponse. Cette mauvaise réponse dans une situation donnée peut être la bonne dans une autre. Le système de classeurs se retrouve alors dans l'incapacité de donner la meilleure solution. Les environnements non Markovien appliqué à des systèmes sans mémoire font partis des problèmes NP-complets [Littman, 1994]. Toutefois [Wilson, 1994] a montré qu'il était possible de transformer tout

environnement non Markovien en un environnement Markovien en élargissant les possibilités sensorielles du système i.e. en ajoutant aux parties condition et action des bits permettant de différencier les situations critiques.

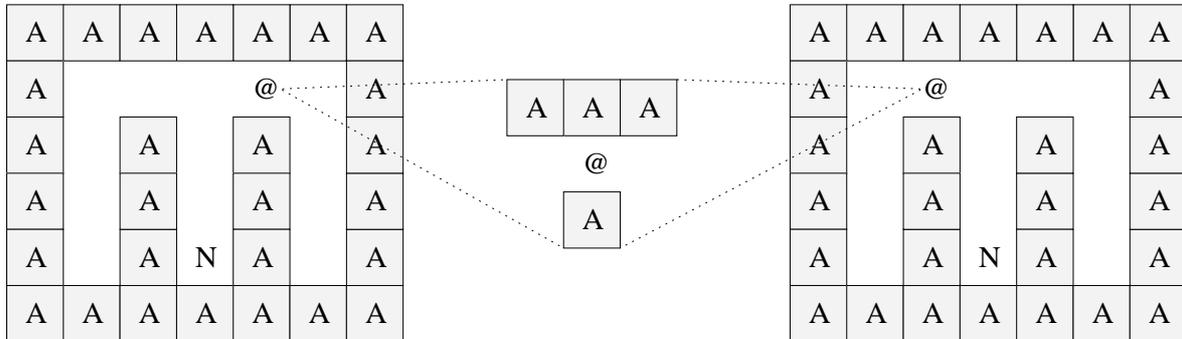


FIG. 3.3 – Le Woods 101 : un environnement non Markovien.

3.1.3 Résolution

Le mécanisme de résolution des systèmes de classeurs est assez simple. L'environnement stimule les capteurs du système. A partir de ceux-ci, un ensemble de classeurs qui reconnaissent le signal est créé. Cet ensemble est appelé Match Set $[M]$. Selon la catégorie de système de classeurs, l'action sera déterminée soit de manière aléatoire en choisissant un des classeurs au hasard. Soit en recherchant la meilleure action à effectuer, ce qui est fait à l'aide de système de prédiction ou d'anticipation pour chaque action disponible dans $[M]$. Une fois l'action déterminée, l'action est retranscrite par les effecteurs à l'environnement qui pourra alors récompenser de manière immédiate ou différée la décision prise. La section 3.3 détaille la particularité de chacun des systèmes de classeurs les plus utilisés.

3.1.4 Évolution

Nous parlerons ici de deux méthodes d'évolution. La première consiste à utiliser un algorithme génétique pour l'évolution. Cet algorithme permet de manipuler les classeurs en les mélangeant de manière Darwinienne, c'est à dire en sélectionnant ceux qui semblent être les meilleurs pour la reproduction. Darwin ne dit pas qu'être le meilleur c'est être le plus fort, il nous rappelle seulement que seuls les êtres les plus adaptés ont survécus. Ces individus sélectionnés, deviennent parfois les parents d'enfants qui peuvent les remplacer quand il n'y a plus de place pour les anciens. Cette méthode permet de mieux parcourir l'espace de recherche et de permettre de faire quelques erreurs dans le choix des individus. C'est effectivement en faisant ses plus belles erreurs que l'homme a fait quelques unes des plus grandes découvertes.

La seconde méthode consiste à corriger le classeur défaillant par l'action attendue. On obtient alors un système élitiste qui ne peut se corriger que si il connaît la solution au problème traité.

3.2 Systèmes de Classeurs avec Algorithme Génétique

Voyons maintenant les caractéristiques propres aux systèmes de classeurs qui utilisent les algorithmes génétiques pour leur évolution. Nous pourrions ainsi en comprendre les atouts puis grâce à l'étude des autres principaux systèmes de classeurs sans algorithme génétique, nous verrons pourquoi nous avons choisi les premiers.

3.2.1 L'algorithme génétique

Introduit par Holland et développé par Goldberg [Goldberg, 1989], les algorithmes génétiques doivent leur nom à des caractéristiques assimilables à celle que la nature utilise au niveau de la cellule. En effet, l'algorithme génétique manipule des chromosomes qu'il sélectionne, croise et mute. Au niveau de la cellule, deux d'entre elles sont sélectionnées naturellement, on les appelle "parents". Ces deux cellules sont croisées afin de créer des cellules "filles". Ce croisement s'opère sur le *génotype* de la cellule i.e. les gènes qui la composent, et manipule les *phénotypes* i.e. ce que représentent les différentes parties du génotype. Lors d'un croisement les deux individus échangent leur capital génétique et participent à l'apparition de nouveaux phénotypes tout en gardant ceux des parents. Lors du croisement de la cellule ou de sa duplication qui sont des opérations chimiques sur les gènes, il arrive parfois qu'un gène et par conséquent, un phénotype, soit modifié par ce que l'on appelle une mutation. Cette mutation s'opère très rarement et risque parfois de créer un phénotype non viable ou inversement un phénotype plus résistant à l'environnement dans lequel la cellule vit. Pour adapter le fonctionnement de la cellule à l'informatique, il a été nécessaire de transformer croisement et mutation en probabilité qu'elle se produise. La génération de nombres aléatoires simule l'évolution naturelle. Les probabilités de croisement et de mutation usuelles ont été déduites et approchées par l'observation de l'évolution. Ainsi, une probabilité de croisement standard entre individus oscillera entre 50% et 100% à chaque génération de l'évolution et une probabilité de mutation standard oscillera quant à elle entre 0.01 % et 0.5 % pour une mutation allélique i.e. par gène et sera multipliée par la taille du chromosome pour une mutation chromosomique. Ces valeurs peuvent être adaptées selon la nature du problème abordé. La nature utilise différentes méthodes pour sélectionner les individus qui vont être croisés. Une forme de sélection naturelle s'opère en essayant de garder les meilleurs individus en fonction de différents critères qui se sont établis au cours de l'évolution. Ce mécanisme de sélection peut être simulé mathématiquement de plusieurs façons.

Mécanisme de sélection

Deux méthodes classiques existent pour simuler la sélection de deux individus qui vont se croiser. Premièrement la sélection par tournoi qui consiste à choisir le meilleur individu et à lui permettre de

se reproduire en éliminant les plus faibles systématiquement. Cette méthode ne laisse rien au hasard et permet de faire dominer rapidement ce qui semble être le meilleur individu dans la population.

La méthode de la roue de la fortune permet d'introduire un peu de hasard en permettant la sélection d'individus totalement différents. Le principe est simple, chaque individu de la population a un poids en terme de fitness. La roue de la fortune consiste à tirer au hasard un nombre compris entre 0 et la somme de toutes les fitness. Ce nombre va permettre de déterminer quel individu va être sélectionné pour le croisement en parcourant la liste des individus dans l'ordre et en accumulant la fitness jusqu'à ce qu'elle soit supérieure ou égale au nombre tiré au hasard. Les individus qui ont les meilleures fitness auront le plus de chance d'être choisis. On peut choisir deux fois le même individu. La table 3.1 nous

Individus	Fitness	Représentativité	Somme des Fitness
1 01001	2	25%	2
2 11110	4	50%	6
3 00001	1	12,5%	7
4 10000	1	12,5%	8

Tirage Roulette	N° Individu Sélectionné	Individu Sélectionné
3	2	11110
7	3	00001
1	1	01001
5	2	11110

TAB. 3.1 – Roue de la Fortune.

montre un exemple de tirage pour une population. La somme des fitness est ici de 8. Lorsqu'on tire 3, c'est l'individu numéro 2 dont la somme des fitness se trouve dans l'intervalle $]2, 6]$ qui est sélectionné. Lorsqu'on tire 7 à la roulette, on sélectionne l'individu 3 car il est celui qui a la valeur 7 lorsqu'on somme dans l'ordre les fitness des individus de la population.

Opérateur de croisement

Au niveau des systèmes de classeurs, cela consiste à choisir deux individus de la population que l'on va croiser pour créer deux nouveaux individus qui soit remplaceront les parents, soit s'ajouteront à la population ou encore remplaceront deux individus moins performants. Un croisement s'opère en un à n points des chromosomes des individus sélectionnés. La figure 3.4 permet de voir le résultat d'un croisement mono-point. Il est facile de constater que la diversité peut être facilement recréée à partir de deux individus croisés. Dans un croisement mono-point, on choisit aléatoirement un endroit identique dans chacun des parents, où le croisement va s'opérer. Tout le capital génétique des parents qui est avant le point de croisement reste identique chez les enfants correspondants, tandis que la partie du chromosome qui se situe après ce point est échangé chez les enfants. On peut obtenir de nouveau les parents en recroisant les enfants au même endroit. Lors d'un croisement n-points, cf. figure 3.5, le brassage est très différent. Les phénotypes sont brassés de manière plus intense. Dans un croisement

n -points, n doit être strictement inférieur à λ , la longueur de la chaîne. Si l'on donne un rang à chaque point de croisement et le rang 0 au début du chromosome, on retrouve dans les allèles des enfants ceux du parent correspondant pour les points situés entre un rang pair et un rang impair et les allèles de l'autre parent pour les points situés entre un rang impair et un rang pair.

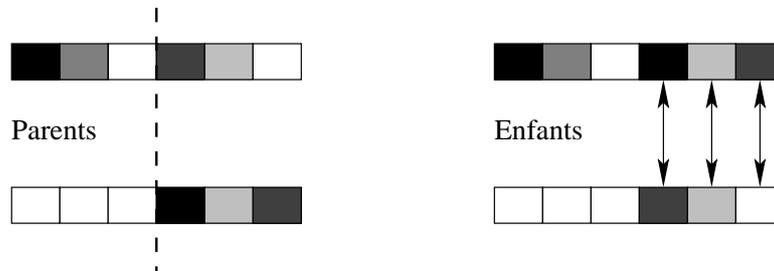


FIG. 3.4 – Croisement mono-point.

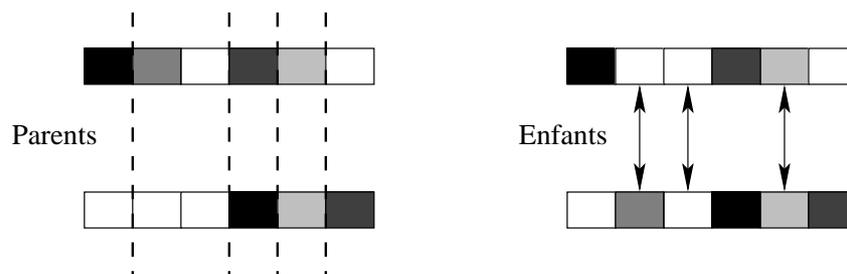


FIG. 3.5 – Croisement n -points.

Opérateur de mutation

Pour un système de classeurs, on effectue deux types de mutation. Soit la mutation allélique qui consiste à faire un tirage aléatoire pour chaque allèle du chromosome ou bien une mutation chromosomique qui consiste à déterminer si il y a une mutation dans ce chromosome. La mutation change un des allèles en un autre. Cela revient à changer la valeur d'un des caractères de la chaîne chromosomique en l'une des autres valeurs possibles. La figure 3.6 montre un des effets possible d'une mutation sur le troisième allèle d'un chromosome.

Convergence

Il existe un théorème sur la convergence d'un algorithme génétique qui indique que l'on est sûr qu'à l'aide d'un opérateur de croisement et de mutation, on obtiendra en un temps non déterminé la solution au problème posé. Alors pourquoi utiliser un algorithme génétique si il faut un temps "infini" pour résoudre un problème ? En fait, puisque l'algorithme génétique intervient pour l'essentiel, au

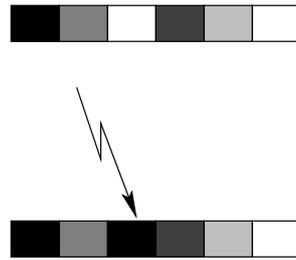


FIG. 3.6 – Mutation.

niveau de l'évolution des systèmes de classeurs, il permet surtout de supposer que la population se stabilise à l'infini autour de la solution du problème. D'autre part, nous verrons que l'algorithme génétique permet d'évaluer plus de 80% des populations d'individus possibles en très peu de générations, ce qui est un atout lorsque le problème rencontré est soit inconnu, soit bruité. Nous avons discuté de la notion de bruit dans le chapitre 2.

3.2.2 Systèmes de classeurs de type Michigan

Ce système de classeurs utilise pour son évolution un algorithme génétique et pour son évaluation un système de récompense appelé *Bucket Brigade* [Holland, 1985]. La bucket brigade, littéralement la brigade des seaux, simule une chaîne de personnes qui se passent les seaux d'eau depuis un point d'eau afin d'éteindre l'incendie d'une maison. L'eau symbolise la récompense qui doit être amenée de personne en personne. Éteindre l'incendie correspond à répondre au problème posé. Ainsi, les membres de la chaîne qui arrivent à éteindre l'incendie sont ceux qui arrivent non seulement à se passer l'eau (la récompense) jusqu'à l'incendie depuis le puits, mais aussi qui le font le plus rapidement i.e. la chaîne la plus courte. Ainsi les personnes deviennent des individus du système de classeurs et l'eau est une récompense qui va aller de membre en membre à chaque génération. Chaque participant aura une récompense quand le dernier de la chaîne aura trouvé l'eau qui est donné par l'environnement. Pour être le premier de la chaîne, il faut être capable de lire le message "au feu" que l'on a reçu du problème. Et pour faire partie de la chaîne solution, il faut être capable d'interpréter le nouveau message que le dernier membre de la chaîne actuelle aura transmis. Enfin pour être le dernier de la chaîne, personne ne doit pouvoir interpréter le message émis. Toutefois, si le message du dernier est mauvais, il n'aura pas d'eau et toute la chaîne qu'il termine risque de disparaître. Inversement si il a eu de l'eau, la chaîne peut subsister. La récompense est donc rétro-propagée au sein des individus. Au niveau du système de classeurs, l'environnement émet un message "au feu" sur un tableau noir qui ne lui coûte rien. Ensuite un individu du système de classeurs qui comprend le signal fait une offre afin de participer à la résolution du problème :

$$Of fre_i = \alpha \times Force_i$$

α est le coefficient d'apprentissage et est compris entre 0 et 1. Si l'individu i est retenu car son offre est soit la meilleure, soit la seule, il doit payer à l'environnement le montant de son offre. Sa force ainsi que celle de l'environnement sont modifiées :

$$Force_i = Force_i - Offre_i$$

$$Force_{env} = Force_{env} + Offre_i$$

Il émet ensuite un nouveau message qui va être interprété par les autres individus qui vont alors répéter le processus en essayant de faire la meilleure offre. Supposons que le classeur j fasse la meilleur offre au classeur i alors :

$$Offre_j = \alpha \times Force_j$$

$$Force_j = Force_j - Offre_j$$

$$Force_i = Force_i + Offre_j$$

Si le message n'est pas interprétable, c'est alors l'environnement qui va recevoir la décision des classeurs et déterminer une sanction :

$$Sanction_{env} = \alpha \times Force_{env}$$

Si le dernier message n'est pas celui attendu :

$$\left\{ \begin{array}{l} Force_{env} = Force_{env} + Sanction_{env} \\ Force_k = Force_k - Sanction_{env} \end{array} \right.$$

Si le dernier message est celui attendu :

$$\left\{ \begin{array}{l} Force_{env} = Force_{env} - Sanction_{env} \\ Force_k = Force_k + Sanction_{env} \end{array} \right.$$

Il peut bien évidemment y avoir plusieurs classeurs qui font la même offre, dans ce cas, tous les classeurs écrivent leur message sur le tableau noir. Ce qui nous amène à l'algorithme 1.

Voyons en détails le mécanisme d'une part de transmission des messages et d'autre part de distribution de la punition / récompense. Dans un premier temps, le signal émis par l'environnement est codé par les *détecteurs* du système de résolution. Le message ainsi reçu est *posté* dans la *liste des messages* appelée également tableau noir. La population de classeurs consulte la liste des messages et certains classeurs seront activés car ils sont capables de déchiffrer le message i.e. leur partie condition correspond, au # près, à l'un des messages postés. Ils vont à leur tour poster un message qui pourra être également lu par d'autres classeurs. Lorsqu'un message n'est pas interprétable par les classeurs, le message est alors émis vers l'environnement à l'aide d'effecteurs. Le message transmis à l'environnement fait l'objet d'une sanction qui sera directement donnée à l'émetteur de ce dernier message.

Algorithme 1 de la bucket brigade.**Début**

Remplir(Class-List) ; { avec tous les classeurs du système. }

Remplir(Mesg-List) ; { avec les messages provenant de l'environnement. }

Remplir(Ancien-List) ; { avec l'émetteur des messages, i.e. l'environnement. }

Répéter**Pour** (Tous les M_j de Mesg-List) **Faire****Pour** (Tous les C_i de Class-List) **Faire**Remplir(Match-List) ; { avec les C_i qui comprennent M_j . }**Pour** (Tous les C_i de Match-List) **Faire**Calculer l'offre : $Offre_i = Force_i \times \alpha$;Remplir (Gagnant-List) ; { avec les C_i de Match-List qui ont fait la meilleure offre. }**Pour** (Tous les C_i de Gagnant-List) **Faire**Calculer la force : $Force_i = Force_i - Offre_i$;**Pour** (Tous les C_j de Ancien-List) **Faire****Pour** (Tous les C_i de Gagnant-List) **Faire**Calculer la force : $Force_j = Force_j + \frac{Offre_i}{Taille(Ancien-List)}$;**Pour** (Tous les M_j de Mesg-List qui n'ont pas été traités) **Faire** $Sanction = \frac{Force_{env} \times \alpha}{Taille(M_j-List)}$; { Calculé une fois. }**Si** (M_j n'est pas le message attendu) **Alors** $Sanction = -Sanction$;Calculer la force : $Force_{env} = Force_{env} - Sanction$;**Pour** (Tous les C_i de Ancien-List ayant émis M_j) **Faire**Calculer la force : $Force_i = Force_i + Sanction$;

Effacer (Mesg-List) ;

Remplir (Mesg-List) ; { avec les actions des C_i de Gagnant-List. }

Effacer (Ancien-List) ;

Remplir (Ancien-List) ; { avec les classeurs de Gagnant-List. }

Effacer (Gagnant-List) ;

Jusqu'à (Mesg-List soit vide) ;**Fin**

Les messages seront par conséquent, tous traités avant de passer à la génération suivante. Les tables 3.2 décrivent l'exécution de l'exemple décrit par la figure 3.7. Pour cet exemple, on fixe $\alpha = 0.1$ et on suppose que l'environnement attend le signal 011 pour offrir une sanction positive. Le premier message est celui émis par l'environnement, à savoir : 001. Au pas numéro un, seul le premier individu est activable par le message de l'environnement. Il émet alors le signal 111 tout en payant une offre égale à $\alpha \times SaForce$ à l'environnement soit 1. Le message émis va déclencher, au pas numéro deux, l'activation des individus 2 et 4 qui vont faire une offre identique à l'individu 1 pour pouvoir émettre à leur tour. Les deux offres sont payées à l'individu 1 qui se retrouve avec une force de 11 et deux nouveaux messages sont émis : 010 et 100. Au pas numéro trois, le message 010 n'est pas interprétable par l'ensemble des classeurs, il est donc émis par les effecteurs auprès de l'environnement. Ce message n'étant pas celui escompté, l'environnement prélève $\alpha \times Force_{env}$ au classeur émetteur (le classeur 2) qui n'a plus qu'une force de 7,9 après cette action. L'environnement a alors 12,1 de force. L'autre

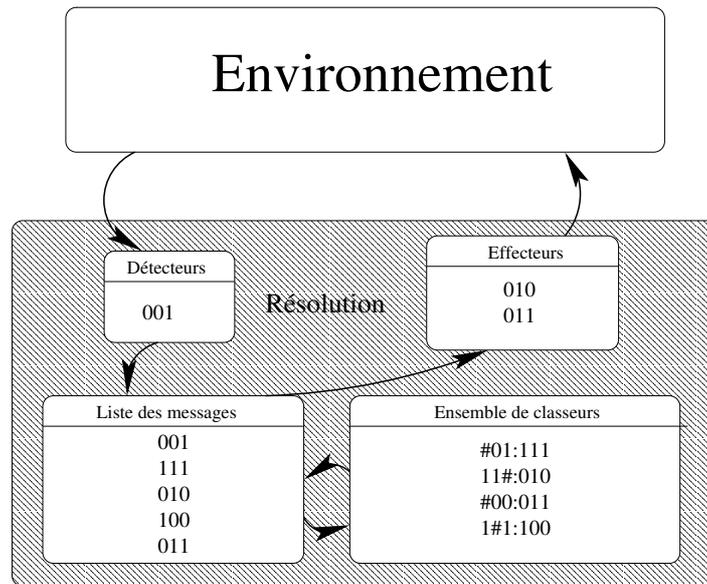


FIG. 3.7 – Mécanisme de résolution de la Bucket Brigade.

message, 100, est traité par le classeur 3 qui émet au pas numéro quatre, le message 011 qui n'est pas interprétable par la population de classeurs. Ce message est émis et donne lieu à une récompense de la part de l'environnement égale à 1,21. L'environnement se retrouve alors avec une force de 10,89.

On voit qu'à la génération suivante, le classeur numéro 4 va commencer à bénéficier de la force gagnée par celui qui est après lui dans la chaîne tandis que le classeur numéro 2 risque de ne pas participer à cette prochaine génération car l'offre qu'il va pouvoir faire devient trop faible.

Pour l'évolution, le système de classeurs de type Michigan utilise un algorithme génétique qui se base sur la force pour brasser les classeurs. En reprenant l'exemple que nous venons de voir et en regardant les forces finales de chacun d'entre eux, on constate que les classeurs ayant le plus de chance d'être éliminé par la sélection sont le numéro 2 dont on sait instinctivement qu'il ne répond pas au problème bien qu'à une mutation d'avoir une récompense, et le numéro 4 qui bien qu'essentiel à la bucket brigade de cet exemple n'a pas encore reçu les bénéfices de sa participation.

La rétro-propagation de la récompense fait courir le risque au système de perdre un individu important de la chaîne qui meurt faute d'avoir eu de l'eau à temps. Ainsi dès que la chaîne qui mène à la solution est trop longue, les systèmes de classeurs de type Michigan se trouvent dans l'impossibilité de trouver une solution stable.

3.2.3 Systèmes de classeurs de type Pittsburgh

Ce système de classeurs introduit par [Smith, 1980] utilise pour son évaluation et son évolution un algorithme génétique, ce qui le différencie beaucoup des autres systèmes de classeurs à algorithme génétique. Pour que ce système puisse fonctionner, un individu représente une solution au problème. Les

Pas 1

Individus	Force ini.	Msg posté	Match msg de	Offre émise	Force fin.
1 #01 :111	10		Environnement	$0.1 \times 10 = 1$	$10 - 1 = 9$
2 11# :010	10				10
3 #00 :011	10				10
4 1#1 :100	10				10
Env.	10	001			$10 + 1 = 11$

Pas 2

Individus	Force ini.	Msg posté	Match msg de	Offre émise	Force fin.
1 #01 :111	9	111			$9 + 2 = 11$
2 11# :010	10		Ind. 1	$0.1 \times 10 = 1$	$10 - 1 = 9$
3 #00 :011	10				10
4 1#1 :100	10		Ind. 1	$0.1 \times 10 = 1$	$10 - 1 = 9$
Env.	11				11

Pas 3

Individus	Force ini.	Msg posté	Match msg de	Offre émise	Force fin.
1 #01 :111	11				11
2 11# :010	9	010			$9 - 1.1 = 7.9$
3 #00 :011	10		Ind. 4	$0.1 \times 10 = 1$	$10 - 1 = 9$
4 1#1 :100	9	100			$9 + 1 = 10$
Env.	11		Ind. 2	$-0.1 \times 11 = -1.1$	$10 - 1.1 = 12.1$

Pas 4

Individus	Force ini.	Msg posté	Match msg de	Offre émise	Force fin.
1 #01 :111	11				11
2 11# :010	7.9				7.9
3 #00 :011	9	011			$9 + 1.21 = 10.21$
4 1#1 :100	10				10
Env.	12.1		Ind. 3	$0.1 \times 12.1 = 1.21$	$12.1 - 1.21 = 10.89$

TAB. 3.2 – Exécution de l'algorithme de la Bucket Brigade.

individus sont donc un ensemble de classeurs qui tente de résoudre le problème posé. Un individu est littéralement une cellule avec son patrimoine génétique composé de plusieurs chromosomes, appelé

également *structure de connaissances*, afin de répondre à l'environnement et nécessite par conséquent que la taille de son patrimoine soit adaptée i.e. ni trop faible, ni trop importante mais suffisante. La taille de ce patrimoine s'exprime par un nombre de classeurs qui compose l'individu, on le nomme également *nombre de règles*. Le nombre de règles contenues dans chaque individu varie au cours du temps. Lors de l'évaluation, les individus qui composent le système de classeurs, sont évalués indépendamment et en parallèle. Un individu dispose d'une force qui lui est donnée par une fonction de fitness appelée originellement "*critique*". Cette force représente la valeur moyenne de chaque classeur qui compose l'individu et est le fruit de k tentatives de résolution du problème abordé. C'est cette même force d'adaptation (de *fitness*) qui va permettre à l'algorithme génétique de brasser les individus entre eux.

Au niveau de son fonctionnement, le système de classeurs de type Pittsburgh se décompose en trois modules [Smith, 1984] :

- Le système de résolution du problème qui utilise les règles qui composent chaque individu pour répondre au problème posé.
- La critique ou fonction de fitness qui mesure la performance de chaque individu aux k essais de résolution du problème.
- Le système d'apprentissage qui est composé d'un algorithme génétique adapté aux individus à taille variable.

La structure des règles qui composent chaque individu peut être plus complexe que celle que nous avons déjà vu. En effet, le mécanisme de résolution prévoit l'utilisation d'un langage complexe ainsi que l'utilisation d'un tableau noir pour poster des messages internes. Chaque règle dispose dans sa partie condition, d'une zone qui sert à capter le signal provenant de l'environnement. Ce signal peut être décomposé en plusieurs signaux représentant différentes variables. Les capteurs de signaux peuvent être complétés d'un bit symbolisé par '-' servant à exprimer la fonction *non*, i.e. la règle est activée si un des bits de la valeur binaire du signal ne correspond pas à la zone de la partie condition concernée. D'autre part, la partie condition de la règle peut manipuler des variables de travail qui peuvent servir au système de messagerie interne ou encore lors de l'action finale. Ces variables sont considérées comme des arguments de la partie condition qui peuvent être instanciées à l'aide de la messagerie interne. La partie action se décompose en une action et un signal à émettre soit dans la messagerie interne, soit de manière externe à faire sur l'environnement. Si les variables de travail ne peuvent être instanciées à l'aide de la messagerie interne, l'action sera alors effectuée sur l'environnement.

Prenons pour exemple ce classeur d'un individu -1##00##0# 1##X 0#0X 001Y \rightarrow 011 *emettre*(Y) et 111101001 le signal provenant de l'environnement.

La table 3.3 contient l'état du tableau noir ou messagerie interne. Le signal de l'environnement permet d'activer le classeur de l'individu. En effet, 1111 correspond parfaitement à *non*(1##0) puisque le dernier bit du signal est différent de celui de la règle. Il en est de même pour 01001 qui correspond à 0##0#. On regarde ensuite comment instancier les variables de la partie condition en se servant de la messagerie interne. 1##X est instanciable par la case 1 de la messagerie interne et rattache la variable

1	100	1111
2	010	
3	110	101010
4	001	101
5	000	1111

TAB. 3.3 – Messagerie interne.

X au signal 1111. La case 5 de la messagerie permet d’instancier $0\#0X$. La variable X a donc deux valeurs possibles. Enfin, la case 4 permet d’instancier la variable Y avec la valeur 101. On peut alors déclencher la partie droite de la règle qui consiste à émettre le signal 011 suivi de la valeur de Y , soit 101, dans la messagerie interne.

Cette fonctionnalité sur les variables n’a pu être implémentée par Smith mais doit permettre théoriquement au système et grâce au parallélisme dans le déclenchement des règles, de définir par communication interne les paramètres idéaux avant d’effectuer une action.

L’algorithme génétique a été modifié afin de prendre en compte la manipulation d’individus de tailles différentes. L’opérateur de croisement doit en effet opérer aussi bien au niveau binaire qu’au niveau des classeurs. Seul un croisement mono-point a été proposé par Smith. On tire aléatoirement un couple n, i pour le croisement du premier parent et un couple n', i pour le second parent avec $0 < n < Taille(Parent_1)$, $0 < n' < Taille(Parent_2)$ et $0 < i < \lambda$ (cf. § 3.1.1). n et n' sont les numéros de classeurs des parents 1 et 2 respectivement où aura lieu le croisement et i est le bit où se produit le croisement dans chaque classeur n et n' concerné. Soit $t = Taille(Parent_1) + Taille(Parent_2)$, la taille totale t' des enfants est égale à t .

Smith a démontré que son opérateur de croisement avait tendance à agrandir les disparités de longueur si les parents avaient une différence de taille trop grande et inversement à réduire cette disparité si les longueurs des parents étaient proches. C’est donc le mécanisme de sélection qui va permettre aux individus de s’agrandir puisqu’on suppose que les individus devenant trop petits vont répondre moins efficacement au problème et auront tendance, par conséquent, à être moins sélectionnés.

Smith utilise également un opérateur d’inversion qui permet de brasser plus fortement les classeurs et permet à son système de converger plus rapidement.

Voyons maintenant en détails un exemple de fonctionnement d’un système de classeurs de type Pittsburgh, à l’aide de la figure 3.8. Une fois la population initialisée aléatoirement, chaque individu va être simulé en parallèle avec un environnement qui va interagir avec lui et sera soit remis à zéro, soit remis dans l’état dans lequel il se trouvait à la génération précédente. Dans ce contexte, nous nous plaçons au début d’une génération, l’environnement parallélisé venant d’être mis à 0.

En fonction de son capital génétique, chaque individu va avoir une réponse au signal émis par l’environnement. Cette réponse fait l’objet d’une sanction qui devient la nouvelle force de fitness de l’individu. Afin que la force reflète bien la valeur moyenne de chaque classeur, il est nécessaire de faire évaluer plusieurs fois chaque individu avant de se servir de cette force pour le faire évoluer. C’est pour

cette raison que l'on évalue un individu à chaque génération d'un nombre de fois k au moins égal à la moitié du nombre de ces règles. Chaque individu sera testé $k = 4$ fois. Par simplification, les individus seront composés d'un nombre identique de classeurs.

Le but du système est de résoudre la fonction booléenne *et* à deux entrées. L'environnement va donc stimuler les individus avec les 4 entrées possibles de la fonction pour vérifier qu'une des structures de connaissances (individu) répond aux critères de cette fonction. Seul le deuxième individu présenté dans la figure 3.8 répond à ce critère. En effet, l'individu 1 émet le bit 1 alors que le message est 00 et ne sera pas récompensé. Le dernier individu présenté quant à lui ne sait pas répondre au signal 00 et ne sera également pas récompensé.

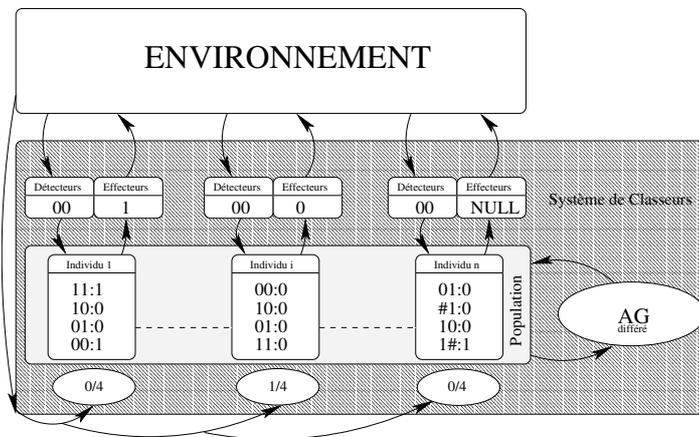


FIG. 3.8 – Exemple d'évaluation d'un système de classeurs de type Pittsburgh.

L'idée de transformer un individu en base de connaissance capable de répondre au problème est l'une des idées majeures du travail de Smith. La complexité induite par la variabilité de la taille des individus empêche toute implémentation utilisant pleinement le potentiel de cette idée. En effet, la taille des individus ne peut pas croître à l'infini et l'inclusion d'un critère de taille à la "critique" entraîne une baisse notable de performance [Smith, 1980].

3.2.4 XCS

Ce système de classeurs, introduit par [Wilson, 1995], utilise pour son évolution un algorithme génétique. Son évaluation se fait grâce à un algorithme évolué de Q-Learning. Le principe est simple et vise à remplacer l'algorithme d'apprentissage de la bucket brigade défaillant par un système de prédiction de récompense des individus [Spiessens, 1990]. Qui dit prédiction, dit erreur de prédiction. L'erreur permet d'évaluer ce que Wilson appelle la perspicacité ("accuracy") d'un système de classeurs. Cette perspicacité sert à la fois au mécanisme d'évolution en permettant de déterminer la nouvelle valeur de l'individu (fitness) et elle permet de savoir si l'individu pourra participer ou non à la prochaine évaluation. Autre particularité du XCS, l'évolution ne s'effectue éventuellement que

dans la niche écologique que représentent les individus ayant déterminé l'action à entreprendre vis à vis de l'environnement.

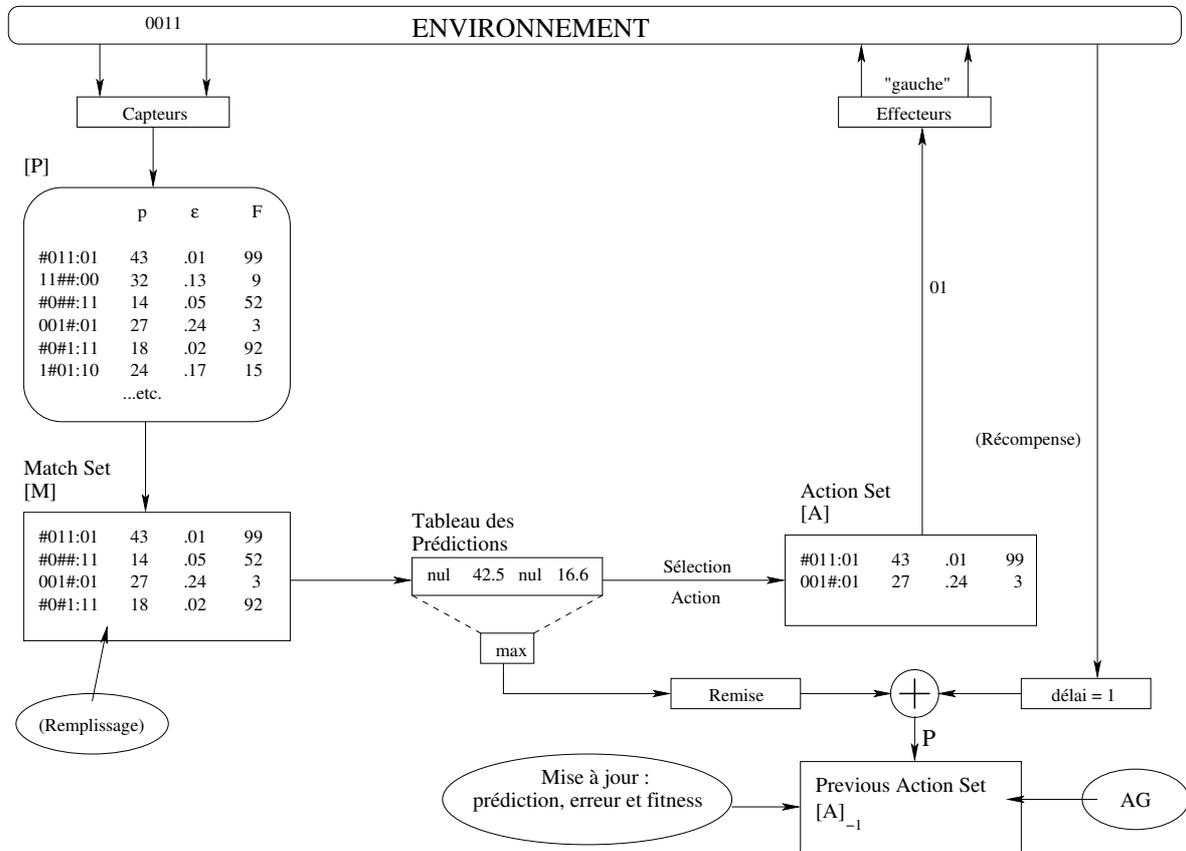


FIG. 3.9 – Illustration du XCS.

La figure 3.9 décrit le fonctionnement du XCS. Les individus sont composés d'un classeur comme pour les systèmes de classeurs Michigan. Les classeurs participant à la résolution vont être sélectionnés par le message provenant de l'environnement. Les classeurs sélectionnés forment alors un *Match Set* [M]. La population initiale pouvant être vide, on a adjoint un système de remplissage qui permet de créer artificiellement des classeurs pour remplir [M]. Les classeurs ainsi générés, auront une partie condition qui correspondra au signal émis par l'environnement à quelques symboles génériques “#” près. Leur partie action est quand à elle remplie aléatoirement. Toutefois, lors de ce remplissage, on doit trouver dans [M] au moins θ_{mna} actions différentes. Une fois le *Match Set* rempli, on va déterminer l'action qui sera choisie à l'aide d'un tableau de prédictions. Pour chaque action possible et disponible dans [M], on va déterminer une valeur qui correspond à la performance prévisible de l'action. La formule est la suivante :

$$P[A] = \frac{\sum p_{[A]} \times F_{[A]}}{\sum F_{[A]}}$$

Il existe deux façons de choisir l'action qui sera appliquée. La première consiste à choisir l'action

qui a la performance prévisible la plus forte. La seconde, par contre, consiste à choisir au hasard une des actions disponibles dans le tableau des prédictions. L'Action Set [A] est alors constitué à partir des classeurs dont la partie action correspond à l'action choisie. Au temps suivant, une récompense est donnée à $[A]_{-1}$. Elle va permettre de mettre à jour la prédiction p , l'erreur de prédiction ε et la fitness F . C'est dans $[A]_{-1}$ qu'aura lieu l'évolution par l'intermédiaire d'un algorithme génétique.

Deux mécanismes ont été ajoutés [Wilson, 1998] afin de permettre au système de contenir une population mieux adaptée :

- Celui de Surgénéralisation qui permet de remplacer un classeur par un autre si celui-ci contient plus de généralité (i.e. de “#”) et s'il a en outre, une prédiction, une erreur et une fitness suffisamment mauvaises et enfin s'il a participé au moins θ_{sub} fois aux décisions.
- Celui d'Elimination par Vote, qui permet d'éliminer les classeurs qui existent depuis longtemps mais qui ne participent jamais aux décisions car leur performance prévisible est trop faible mais aussi qui ont participé au moins θ_{del} fois aux décisions.

Afin d'éviter de surcharger la population d'individus doublons, [Butz et Wilson, 2000] a eu l'idée de regrouper les classeurs en leur donnant une quantification et une date de naissance ainsi qu'un âge comme nous venons de le voir avec les paramètres θ_{sub} et θ_{del} . Nous obtenons un système à base de Q-Learning (cf. § 3.3.1) pour la mise à jour de la force de fitness :

$$\begin{aligned}
 p_j &= p_j + \beta \times (P - p_j) \\
 \varepsilon_j &= \varepsilon_j + \beta \times (|P - p_j| - \varepsilon_j) \\
 \begin{cases} \varepsilon_j > \varepsilon_0, \kappa_j = \exp[(\ln(\alpha)) \times (\varepsilon_j - \varepsilon_0)] \\ \varepsilon_j \leq \varepsilon_0, \kappa_j = 1 \end{cases} \\
 F_j &= F_j + \beta \times \left(\frac{\kappa_j}{\sum_{i \in [A]} \kappa_i} - F_j \right)
 \end{aligned}$$

où j est l'indice d'un classeur membre de [A].

Les classeurs qui ont participé à l'action peuvent éventuellement évoluer à l'aide d'un algorithme génétique si ils sont âgés d'au moins θ_{GA} participations aux décisions depuis leur naissance ou leur dernière évolution.

Le XCS a fait ses preuves en environnement Markovien [Butz et Pelikan, 2001]. Lorsqu'on l'adapte aux environnements non-Markovien par la méthode de Wilson, le système réagit bien [Lanzi, 1997, Lanzi et Wilson, 2000, Butz et al., 2001]. Cet outil est également une référence dans le domaine des systèmes de classeurs à algorithme génétique.

3.3 Autres Systèmes de Classeurs

Nous avons vu les principaux systèmes de classeurs avec algorithme génétique. Cette section va nous permettre de faire un tour rapide des autres systèmes de classeurs.

3.3.1 Q-Learning

Le Q-Learning introduit par [Watkins, 1989], est une méthode d'apprentissage qui permet de faire évoluer un système d'apprentissage en évaluant la performance de toutes les actions possibles puis en choisissant la meilleure. De manière formelle :

- soit E l'ensemble des états possibles du système.
- soit A l'ensemble des actions possibles.
- soit β et γ les facteurs d'apprentissage.
- soit R la récompense associée à un état $e \in E$ et à une action $a \in A$.
- soit $e' \in E$ l'état résultant de l'application du couple (e, a) et $a' \in A$ la meilleure action dans l'état e' .

Alors :

$$Q(e, a) \leftarrow (1 - \beta) \times Q(e, a) + \beta \times (R + \gamma \times Q(e', a'))$$

Nous obtenons également un tableau des prédictions $P \in [0, 1]$ qui pour chaque triplé (e, e', a) indique la probabilité de choisir a comme action dans l'état e :

$$P(e, e', a) = \frac{Q(e, a)}{\sum_{a'' \in A} Q(e', a'')}$$

Utilisons la notation francisée de [Dorigo et Bersini, 1994] pour adapter le Q-Learning aux systèmes de classeurs :

- E , l'ensemble des états, devient C , l'ensemble des conditions possibles (cf. § 3.1.1) et c_i , le $i^{\text{ème}}$ classeur $c \in C$.
- β et γ , les facteurs d'apprentissage sont remplacés par α .
- $F_t()$ est la fonction qui calcule la mise à jour de la force du classeur i à l'instant t .

La formule devient alors :

$$F_{t+1}(c_i, a_i) = (1 - \alpha) \times F_t(c_i, a_i) + \alpha \times (R + F_{t+1}(c_j, a_j))$$

Pour être maximale l'activation du classeur i entraîne l'activation du classeur j au pas suivant. Cet algorithme est exactement celui de la Bucket Brigade (§3.2.2).

Nous constatons que cet algorithme ne fonctionne que pour des triplés (e, e', a) , dans le cas général, et (c_i, c_j, a_i) , dans le cas des systèmes de classeurs, où i et j sont uniques i.e. e' est unique pour un e et un a donné. En effet, dans le cas contraire, on ne peut pas maximiser la récompense car nous ne sommes plus dans un système déterministe. L'algorithme d'apprentissage du Q-Learning ne peut fonctionner idéalement qu'avec un environnement Markovien. Si tel est le cas, le Q-Learning offre un algorithme relativement coûteux mais très performant. La plupart des jeux d'échecs utilisent ce genre d'algorithme pour savoir quel coup jouer. Seul le temps de calcul permet d'atteindre une profondeur suffisante pour essayer de battre son adversaire. La faiblesse de ce système est donc sa fonction de maximisation de la récompense qui engendre l'incapacité du système à disposer d'une

stratégie à moyen et long terme quel que soit le type de problème. La perfection seule n'est donc a priori pas un bon choix pour l'apprentissage d'un problème complexe [Sen et Sekaran, 1995] même si le Q-Learning permet d'obtenir de très bons résultats en environnement Markovien. C'est à partir de ce constat que Wolfgang Stolzmann a essayé d'aider le système à connaître l'avenir afin d'anticiper la situation non-déterministe dans laquelle il se trouve.

3.3.2 Anticipatory Classifier System (ACS)

L'ACS présenté par Wolfgang Stolzmann [Stolzmann, 1998] a pour ambition d'étendre le Q-Learning en lui permettant de ne pas être bloqué par les environnements non markovien. L'idée est simple :

- Soit C l'ensemble des conditions possibles du système de classeurs.
- Soit $c_t \in C$, la partie condition activée à l'instant t .
- Soit l'action $a \in A$, l'ensemble des actions possibles.

Un classeur de l'ACS est alors composé du triplé :

$$(c_t, c_{t+1}, a)$$

Le système de classeurs fonctionne toujours à l'aide du couple (c_t, a) . C'est au niveau du système de récompense que l'on va voir la différence. En effet, la force F du système de classeurs est mise à jour à l'aide d'un algorithme de Q-Learning si :

- $Match(Env_{t+1}, c_{t+1}) \wedge (Env_{t+1} \neq Env_t) \rightarrow F(c_t, c_{t+1}, a) = (1 - \beta^{ant}) \times F(c_t, c_{t+1}, a) + \beta^{ant}$
- $Match(Env_{t+1}, c_{t+1}) \wedge (Env_{t+1} = Env_t) \rightarrow F(c_t, c_{t+1}, a) = (1 - \beta^{ant}) \times F(c_t, c_{t+1}, a)$

β^{ant} est le coefficient d'apprentissage de l'anticipation. On diminue la force des classeurs qui ne changent pas l'environnement et inversement on augmente celle de ceux qui ont transformé l'environnement par leur action. Si c_{t+1} ne permet pas partiellement de s'activer à l'aide du signal de l'environnement, on modifie alors le classeur (c_t, c_{t+1}, a) de telle sorte que les parties de c_{t+1} qui ne correspondent pas soient remplacées par des # (cf. § 3.1.1). Si c_{t+1} ne peut pas être partiellement corrigé alors on diminue la force du classeur comme pour les classeurs qui ne changent pas l'environnement.

Le fait de vérifier que le classeur choisi va amener le système dans un état connu permet de résoudre les problèmes non-Markovien à un état non-déterministe. Si la chaîne solution doit être plus longue, l'ACS se trouve alors en difficulté. Toutefois il faut reconnaître que l'idée est bonne et permet au système réactif d'anticiper la prochaine action. Depuis l'ACS, de nombreuses améliorations ont été apportés. En effet, on peut se demander si en modifiant le système de récompense, on ne peut pas améliorer les performances. Toutefois, le coût en espace n'est pas négligeable et on imagine difficilement d'utiliser l'ACS dans un problème où il y a plus d'une situation non-déterministe dans la chaîne solution.

3.3.3 Systèmes de classeurs flous.

Les systèmes de classeurs dit flous, sont des systèmes de classeurs qui au lieu de trancher les réponses au problème par des valeurs entières, propose d'utiliser des fonctions à valeurs réelles [Bonarini, 2000]. Ces fonctions sont composées de variables qui sont identifiées par les parties conditions et actions des classeurs. Il s'agit donc d'un encodage des variables. Les x premiers bits donnent le numéro de la variable concernée. Les y suivants donnent la valeur ou le groupe de valeurs auquel appartient la variable. Il peut y avoir plusieurs variables encodées dans la partie condition. La partie action permet de faire la même chose en encodant la variable sur x bits et sa valeur ou groupe de valeurs sur y bits. Cette technique peut s'appliquer à n'importe quelle catégorie de système de classeurs, avec ou sans algorithme génétique. Prenons un exemple [Valenzuela-Rendón, 1998a] avec deux fonctions *chaud* et *froid* qui indiquent si il fait chaud ou froid :

$$\mu_{chaud}(t) = \begin{cases} \frac{4e^{-(t-40)/10^{\circ}C}}{(1+e^{-(t-40)/10^{\circ}C})^2}, & \text{si } t < 40^{\circ}C \\ 1, & \text{si } t \geq 40^{\circ}C \end{cases}$$

$$\mu_{froid}(t) = \begin{cases} 1, & \text{si } t \leq 0^{\circ}C \\ \frac{4e^{-t/10^{\circ}C}}{(1+e^{-t/10^{\circ}C})^2}, & \text{si } t > 0^{\circ}C \end{cases}$$

On pourra imaginer comme exemple de codage :

0 0001 : 0 01 ou 0 1000 : 0 10

Le premier 0 indique la variable concernée, ici t . Les 4 bits suivants indiquent une classe de température à laquelle appartient t . La classe 0001 indique une température $\leq 0^{\circ}C$. La classe suivante (0010) est de 0 à $20^{\circ}C$, la suivante (0100) de 20 à $40^{\circ}C$ et enfin la dernière (1000) concerne les plus de $40^{\circ}C$. Enfin dans la partie action le premier bit indique également la variable concernée et les deux bits suivants à laquelle des deux fonctions correspond la variable. Ici, 01 indique μ_{froid} et 10 correspond à μ_{chaud} . Donc les deux classeurs indiquent que :

Pour le premier, si $t \leq 0^{\circ}C$ alors $t \in \mu_{froid}$

Pour le second, si $t \geq 40^{\circ}C$ alors $t \in \mu_{chaud}$

L'utilisation de systèmes de classeurs flous impose de connaître à l'avance les intervalles de valeurs sur lesquelles s'appliquent les fonctions de la partie action [Fogarty, Carse et Bull, 1995]. Ils ont un grand pouvoir de synthèse mais ne permettent pas d'aborder facilement un problème méconnu.

3.4 Conclusion

Nous avons décrit dans ce chapitre les principaux systèmes de classeurs [Lanzi et Riolo, 2000] et leur structure. Nous avons pu voir dans un premier temps les traits communs à tous les systèmes de classeurs. Nous avons pu ainsi dégager plusieurs caractéristiques essentielles comme la structure pyramidale de la population qui compose un système de classeurs, l'interaction nécessaire avec un environnement pour apprendre, la structure des classeurs (des règles de production composées d'une partie condition et d'une partie action) et enfin le mécanisme d'évolution directement lié à l'appren-

tissage résultant de l'interaction avec l'environnement. Enfin nous avons revisité les deux grandes catégories de systèmes de classeurs, avec et sans algorithme génétique. Les méthodes automatiques i.e. sans algorithme génétique, font actuellement l'objet d'évolutions intéressantes. Toutefois, nous nous sommes attachés à étudier les systèmes de classeurs à algorithme génétique pour leur propriété adaptative en milieu dynamique bruité. Nous allons appliquer lors des prochains chapitres, un système purement phylogénique (les systèmes de classeurs à algorithme génétique) à un problème ontogénique : communiquer au sein des systèmes multi-agents [Turney, 1996]. Nous étudierons dans le chapitre 4 plus particulièrement les caractéristiques d'une version simplifiée du système de classeurs de type Pittsburgh lors d'une étude menée sur la non communication.

Chapitre 4

Coordination “spontanée”

Nous l’avons vu, dans la nature, le phénomène de coordination “spontanée” peut être effectué par des agents qui ne sont pas autonomes en permanence. La coordination semble avoir ses limites. Nous allons voir quelques définitions inhérentes aux propriétés des agents autonomes tentant de se coordonner. En étudiant un problème de coordination spontanée, nous allons pouvoir étudier la dynamique des agents et ainsi mieux comprendre comment les utiliser. Enfin nous établirons sous la forme d’un bilan, le lien entre notre étude et les propriétés des agents autonomes.

4.1 Définitions

Les agents autonomes se coordonnent de manière spontanée lorsque leur survie en dépend (cf. chapitre 2), à savoir quand la fonction de récompense qui les guide les amène à prendre position pour à la fois survivre et indirectement, permettre aux autres agents de survivre. Voyons quelques concepts connus à propos des agents rationnels :

1. Le Choix (C) : C’est la capacité d’action d’un agent. C’est le potentiel de décision de la partie action des classeurs.
2. La Connaissance (K) : C’est le nombre de ressources visibles par l’agent, c’est à dire le nombre de phénotypes ou variables, qui composent la partie condition des classeurs.
3. L’inertie (I) : C’est la tendance qu’a un agent à rester dans la même fenêtre de choix même si une meilleure alternative est connue. C’est donc le choix d’une stratégie de comportement de la part de l’agent.
4. Le partage (S) : C’est ce que les agents partagent comme connaissance. Cela peut être une information sur ce que font les autres agents dans leur totalité ou non.
5. La précision (P) : On la mesure par l’imprécision qui est la distance que l’on autorise aux agents à la coordination idéale. Un système qui optimise souhaite une précision de 100%.

L’ensemble de ces propriétés d’un agent peut varier au cours du temps. [Rustogi et Singh, 1999] ont montré qu’avoir une connaissance K plus importante que le nombre de choix C n’apporte rien, ce qui

paraît naturel. Par contre le nombre de pas à la coordination du système semble proportionnel à $C \times K$ quand $K \leq C$. L'imprécision permet quant à elle aux systèmes plus complexes de se coordonner plus rapidement. Une forte inertie permet au système de se coordonner rapidement malgré un nombre de choix plus important et une connaissance plus grande. Les algorithmes génétiques semblent rentrer dans la catégorie des agents avec une forte inertie et une grande précision puisqu'ils ont une tendance naturelle à converger rapidement et ont quelques difficultés à reconverger selon leur utilisation.

La figure 4.1, qui provient du travail de [Rustogi et Singh, 1999], permet de mieux comprendre le comportement général d'un système multi-agents en fonction de deux caractéristiques de ses agents. Ainsi, les agents appartenant à la région *I* ont une faible inertie et une faible imprécision. Ils sont alors dépendants du partage de connaissance ce qui implique qu'une connaissance locale et peu de choix permettent une meilleure coordination.

Les agents de la région *IIa* ont une inertie modérée. Leur donner plus de choix ou plus de connaissance n'est pas une priorité pour améliorer la coordination.

Les agents de la région *IIb* sont peu précis, ils ont donc une grande latitude pour se coordonner. C'est pour cette raison que leur donner plus de possibilités d'action ou plus d'informations ne leur permet pas de mieux se coordonner.

Enfin les agents qui sont dans la région *III* ont une telle inertie que le système multi-agents devient inactif i.e. rien ne se passe au niveau du choix des agents.

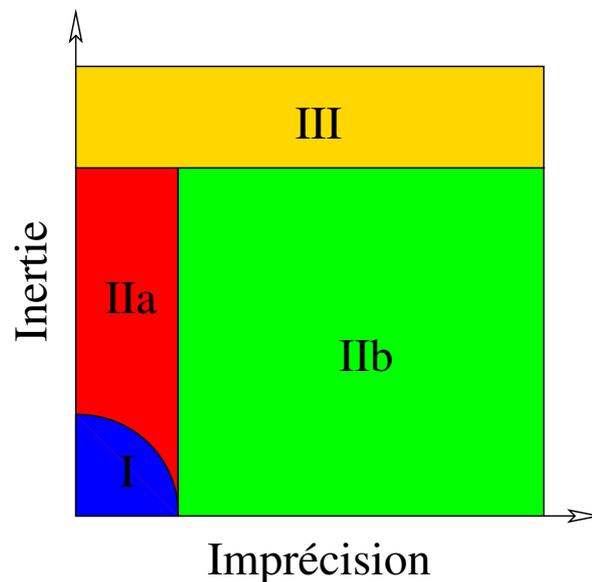


FIG. 4.1 – Comportement typique d'un SMA autonome en fonction de ses agents.

Nous verrons que cette façon de classer les agents qui se coordonnent nous permettra de mieux comprendre l'agent autonome que nous allons présenter dans les prochaines sections.

4.2 El-Farol Problem

Le problème du bar 'El-Farol' a été introduit par [Arthur, 1994]. Il s'agit pour des étudiants d'essayer d'aller dans le bar à la mode 'El-Farol' de l'université de Santa-Fe pour y passer un moment agréable. Derrière cette définition simple se cache de nombreux points à éclaircir. Tout d'abord, comment doit-on considérer l'agent "étudiant", ensuite qu'est-ce que signifie passer un moment agréable et enfin de quelles informations disposent les agents pour décider si en allant au bar ils vont passer un moment agréable.

On suppose que l'agent étudiant est rationnel, c'est à dire qu'il prend ses décisions en fonction des informations dont il dispose et de façon à ce qu'il passe un bon moment. Il a également une mémoire, c'est à dire une capacité à se souvenir des décisions qui ont pu lui servir par le passé. Il doit également pouvoir changer d'avis.

Passer un bon moment dans le bar 'El-Farol' signifie simplement qu'il y a un seuil d'affluence au bar au delà duquel chaque agent aura l'impression de ne pas passer un bon moment et ceux qui ne sont pas venus seront satisfaits de leur décision. Inversement, si le seuil n'est pas dépassé, les agents étant restés chez eux, vont regretter leur choix tandis que ceux qui seront venus seront satisfaits. Les agents ont deux choix C possibles : aller ou ne pas aller au bar. Nous appelons *équilibre* le seuil d'affluence au bar.

Afin de prendre leur décision, les agents ne disposent que d'une seule information, le nombre d'agents qu'il y avait la semaine précédente dans le bar. La connaissance K des agents est donc la même que celle partagée S . L'information dont chaque agent dispose est la même pour tous, c'est la variable d'affluence au bar ($K = 1$). Elle ne donne pas de détails sur la décision de chacun des agents mais permet de connaître l'ensemble des décisions : $K = S = 1$. Les agents appartiennent à la région I de la figure 4.1.

Les agents ainsi définis, sont autonomes, ils ne communiquent pas avec les autres agents. Toutefois ils doivent se coordonner pour avoir le maximum de récompense puisque la récompense dépend du nombre d'agents qui sont allés au bar.

Nous avons voulu rendre ce problème dynamique en changeant la valeur d'équilibre à intervalles réguliers ainsi que le nombre de ces équilibres, afin de voir le comportement des agents évoluer et atteindre ses limites de coordination. La section suivante présente les résultats que nous avons obtenus.

4.3 Expérimentations

La population des étudiants du bar est représentée par 127 agents qui doivent apprendre à aller au bar en fonction d'un nombre idéal de personnes dans le bar qui varie de 32 à 80 agents par pas de 2 à 24 agents et de manière cyclique.

4.3.1 Fonction de fitness

La fonction de récompense des agents est donc dépendante de l’affluence idéale au bar. La fonction de récompense se résume ainsi :

$$\left\{ \begin{array}{l} Af > Eq \left\{ \begin{array}{l} \text{Agent va au bar} \Rightarrow \frac{\text{NombreAgents} - Af}{\text{NombreAgents} - Eq} \\ \text{Agent reste chez lui} \Rightarrow 1.0 \end{array} \right. \\ \\ Af \leq Eq \left\{ \begin{array}{l} \text{Agent va au bar} \Rightarrow 1.0 \\ \text{Agent reste chez lui} \Rightarrow \frac{Af}{Eq} \end{array} \right. \end{array} \right.$$

avec Af l’affluence réelle au bar à la fin d’une évaluation, Eq l’affluence idéale dans le bar i.e. l’équilibre à atteindre, NombreAgents le nombre total d’agents qui participent à l’évaluation. La figure 4.2 montre cette fonction avec une affluence idéale (Eq) de 48 agents et un nombre total d’agents égal à 127. L’axe des abscisses indique l’affluence mesurée pour laquelle correspond dans l’axe des ordonnées, la récompense en fonction de la décision des agents d’aller ou non au bar.

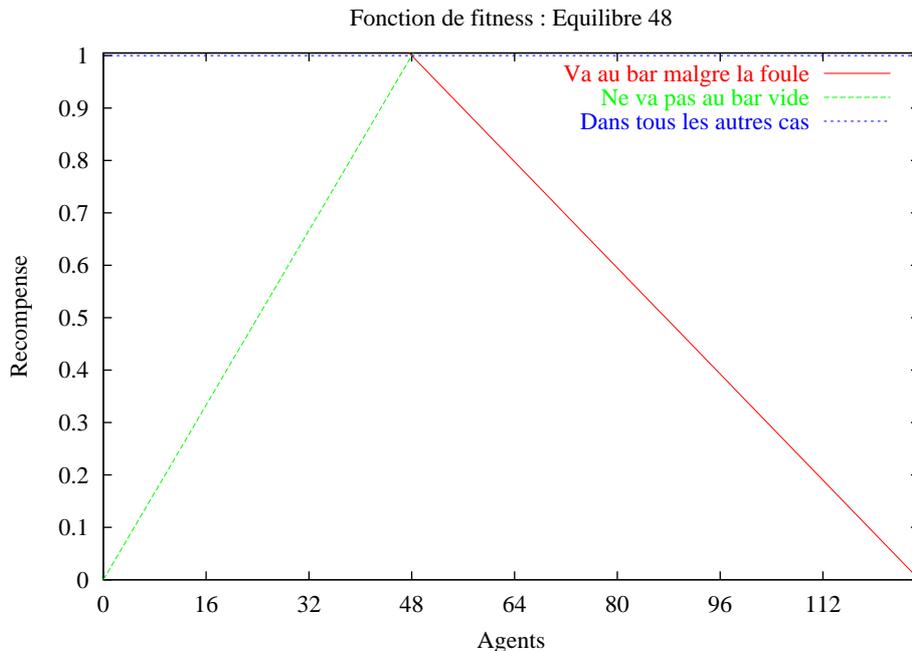


FIG. 4.2 – Fonction de fitness.

Chaque agent est constitué d’un système de classeurs de type Pittsburgh simplifié (cf. § 4.3.3) qui va prendre sa décision en fonction de l’affluence lors de la précédente évaluation. Pour représenter les 128 valeurs possibles d’affluence, la partie condition de chaque classeur est composée de 7 bits ternaires. La partie action consiste en 1 bit qui indique si l’agent va au bar (*vrai* ou *faux*).

Même si les agents sont physiquement homogènes, ils restent indépendants les uns des autres dans la prise de décision. Ce sont des agents autonomes. Il convient également de rappeler qu’étant composés de plusieurs individus, les agents vont se retrouver successivement dans les différents individus

qui composent le système de classeurs les représentant (cf. § 4.3.3). L'environnement, à savoir l'affluence au bar, est mémorisée pour chaque groupe d'individus évalués, afin d'avoir un environnement endogène (cf. § 3.2.3).

4.3.2 Mesures

Nous avons utilisé des paramètres "standards" pour l'algorithme génétique afin de n'étudier que les agents eux-même et leur comportement en terme de coordination. Nous avons successivement étudié l'influence des changements d'équilibre, du nombre d'individus et du nombre de règles. Pour appuyer cette étude, nous avons mesuré l'affluence et l'absentéisme au bar, l'uniformité des individus, la capacité de mémorisation et la part de non-déterminisme des individus.

L'influence de la fréquence des changements d'équilibre. La fréquence des changements d'équilibre passe de 50 à 100 puis 150 et enfin 250 générations. Le but de ces changements est de faire apparaître les limites à la capacité de résolution du problème lorsqu'on limite le temps alloué aux agents pour trouver le seuil d'équilibre. Les quatre équilibres différents changent de manière cyclique et séquentielle de 32 à 80 agents pour cette étude.

L'influence du nombre de changements d'équilibre. Changer le nombre d'équilibre permet de définir une autre limitation à la coordination du système multi-agents. Plus il y a de changements d'équilibre, plus le système doit s'adapter rapidement et doit avoir de possibilités de mémorisation des différentes situations rencontrées. La force d'adaptation du système permet de connaître son inertie. Plus il s'adapte rapidement moins son inertie est grande.

L'influence du changement du nombre d'individus par systèmes de classeurs. Le nombre d'individus a été réglé à 6, 12 et 18. La variation de ce paramètre doit permettre de mieux cerner son rôle dans le comportement du système multi-agents et dans le système de classeurs. Les individus qui composent chaque système de classeurs, représentent un répertoire de comportement comme le souligne [Samuels et al., 2001] :

Evolutionary cognitivists defend a modular conception of mental architecture which views the mind as composed of a large number of special purpose information processing "modules" that have been shaped by natural selection to handle the sorts of recurrent information processing problems.¹

L'évaluation d'un agent consiste donc à évaluer différents répertoires de comportements possibles, c'est à dire à évaluer chaque individu qui le compose. Afin d'estimer l'influence de ce paramètre, nous avons décidé de figer le nombre d'individus qui composent l'agent.

¹Les cognitivistes évolutionnaires défendent une conception modulaire de l'architecture mentale qui considère l'esprit comme une composition d'un grand nombre de modules dédiés au traitement de certaines informations, modules qui ont été façonnés par la sélection naturelle pour résoudre les différents problèmes de traitement des informations récurrentes.

L’influence du changement du nombre de règles par individu. De même que pour le nombre d’individus, le nombre de règles a été réglé à 6, 12 et 18. C’est également pour mieux comprendre le rôle de ce paramètre que nous avons fixé sa valeur. En effet, les règles de production qui composent chaque individu forment la structure de connaissances de celui-ci et représentent par conséquent les briques de base du phénomène d’organisation, y compris social [Hayek, 1992]². En figeant le nombre de règles, nous permettons de mesurer l’influence de ce paramètre sur l’organisation des agents autour de l’équilibre à atteindre.

L’uniformité de la population. L’uniformité de la population est une mesure de la ressemblance moyenne des individus constituant un système de classeurs. Cette mesure se justifie car les individus qui composent chaque système de classeurs ont le même nombre de règles et sont brassés par un algorithme génétique qui va avoir tendance à faire dominer l’individu le plus adapté grâce aux mécanismes de sélection et de reproduction pour l’essentiel. La ressemblance consiste à comparer les règles de même rang des différents individus qui composent les agents et de comparer les bits qui composent les règles. Lorsqu’un caractère # remplace une valeur binaire, le bit est considéré comme similaire à la règle de référence. L’uniformité d’une population permet de connaître sa capacité d’inertie du point de vue évolutionnaire. D’autre part, nous pourrions étudier l’émergence d’un “super-individu” solution dans la population. Cette mesure est possible car nous n’utilisons pas l’opérateur d’inversion de l’algorithme génétique (cf. § 4.3.3).

La capacité de mémorisation. La capacité de mémorisation est exprimée par la capacité d’un agent à revenir à une situation antérieure, i.e. à choisir une stratégie. Dans le cadre du problème du bar, il s’agit de constater si un agent garde la même stratégie pour un équilibre à atteindre donné. Pour ce faire, on comptabilise son comportement dominant pour un équilibre donné et on regarde si il est le même lors de la phase suivante de même équilibre. Nous avons choisi de mesurer la capacité de mémorisation des agents car comme le dit [Chattoe, 1998] :

Agents are extremely good at remembering strategies which they have already implemented or observed elsewhere.³

La part de non-déterminisme dans la prise de décision. Comme [Hayek, 1952] l’explique, deux types d’ordres sont possibles dans l’organisation mentale des agents :

1. *Les organisations.* Elles sont le fruit d’une conception humaine délibérée. Une organisation représente donc toutes les sortes de constructions sociales collectives et individuelles qui reposent sur des règles orientées dont les objectifs sont fixés par une autorité [Chavance, 2001].
2. *L’ordre spontané.* Il est le résultat d’un arrangement involontaire. Un ordre spontané est par conséquent basé sur des règles abstraites dont la nature est, pour l’essentiel, évolutionnaire.

²Pages 20 à 23.

³Les agents sont très efficaces lorsqu’il s’agit de se souvenir de stratégies qu’ils ont déjà acquises ou observées ailleurs.

Ces règles sont en quelque sorte, libres de tout objectif lorsqu'elles apparaissent, se diffusent et survivent grâce à leur capacité d'adaptation.

Pour mémoire, la prise de décision s'effectue grâce aux règles de production qui composent l'individu. Il arrive que pour une même valeur d'entrée, l'agent puisse prendre plus d'une décision. Cette valeur, qui est une affluence la semaine précédente, est considérée alors comme amenant à un choix non-déterministe. La part de non-déterministe consiste donc à évaluer le nombre de valeurs d'entrée qui mènent à un choix non-déterministe par rapport au nombre totale de valeurs d'entrée possibles (ici 128). La part de non-déterminisme reflète l'émergence d'un ordre spontané au sens de Hayek tandis que les mesures concrètes d'affluence et d'absentéisme au bar représentent le choix organisationnel des agents afin de répondre au problème.

Présentation des résultats Nous avons présenté l'évolution des résultats ($V_{Actuelle}$) en calculant le gain (%) en distance à la valeur idéale d'affluence au bar (Eq) par rapport à la situation présentée précédemment ou de référence ($V_{Reference}$). Ce qui s'exprime par la formule suivante :

$$Gain = \frac{(V_{Actuelle} - V_{Reference})}{(Eq - V_{Reference})} * 100$$

Cette mesure permet de visualiser uniformément l'effort qu'a fourni le système pour se rapprocher de la valeur recherchée (Eq).

4.3.3 Pittsburgh Simplifié

Nous avons préféré étudier les systèmes de classeurs de type Pittsburgh à ceux de type Michigan suite à l'étude de [Baiardi et al., 1995]. La version simplifiée d'un système de classeurs de type Pittsburgh consiste à figer le nombre des règles qui composent chaque individu. On obtient ainsi des individus de taille fixe qui évoluent grâce à un algorithme génétique non modifié. Les opérateurs de sélection et de croisement ne nécessitent plus d'adaptation pour préserver les individus performants. Comme nous l'avons indiqué dans la section 4.3.2, figer le nombre des règles qui composent chaque individu, permet également de mieux comprendre l'influence de la taille de chaque structure de connaissances, i.e. individu, sur la résolution de problème par le système de classeurs.

Comme l'évaluation d'un individu s'effectue sur k essais, la fonction de récompense du Pittsburgh simplifié peut renvoyer en fonction du problème traité, une récompense pour chaque essai ou pour la totalité des essais d'une évaluation. La récompense est réinitialisée à chaque nouvelle évaluation.

L'exemple d'évaluation détaillé dans les tables 4.1 utilise le dernier individu représenté dans la figure 3.8. Il s'agit également de résoudre la fonction booléenne $et : s = e_1 \wedge e_2$. Pour tester toutes les réponses possibles, $k = 4$, et durant l'évaluation, l'environnement émet les 4 entrées possibles de la fonction booléenne et à deux entrées.

Essai 1

Message Environnement	00
-----------------------	----

Classeurs		Activables	Activé
1	01 :0		
2	#1 :0		
3	10 :0		
4	1# :1		

Action Choisie	Aucune
Nouvelle fitness	0/4

Essai 2

Message Environnement	01
-----------------------	----

Classeurs		Activables	Activé
1	01 :0	x	x
2	#1 :0	x	
3	10 :0		
4	1# :1		

Action Choisie	0
Nouvelle fitness	1/4

Essai 3

Message Environnement	10
-----------------------	----

Classeurs		Activables	Activé
1	01 :0		
2	#1 :0		
3	10 :0	x	x
4	1# :1	x	

Action Choisie	0
Nouvelle fitness	1/2

Essai 4

Message Environnement	11
-----------------------	----

Classeurs		Activables	Activé
1	01 :0		
2	#1 :0	x	x
3	10 :0		
4	1# :1	x	

Action Choisie	0
Nouvelle fitness	1/2

TAB. 4.1 – Exemple du Pittsburgh.

Le premier signal reçu par l'individu évalué est 00. Il ne permet l'activation d'aucun classeur de l'individu. L'environnement ne donne alors aucune récompense à l'individu qui ne peut pas répondre. Lors du deuxième essai, l'environnement envoie le signal 01. Il permet d'activer deux classeurs de l'individu. Il existe alors plusieurs méthodes pour choisir : soit on prend un classeur au hasard parmi ceux qui sont activables, soit on prend celui qui est le plus spécifique (i.e. celui avec le moins de "#"), soit encore celui qui est le plus général (i.e. celui avec le plus de "#"). Il existe d'autres méthodes que nous ne décrivons pas ici. Dans cet exemple nous utiliserons le choix du classeur le plus spécifique. Ainsi donc le classeur activé est le classeur numéro 1. Il émet alors 0 vers l'environnement qui sanctionne positivement l'individu. Cette sanction indique un essai réussi, sa valeur est donc augmentée de 1 divisé par le nombre d'essais entre deux générations, ici 4. Sachant que la force d'un individu est remise à zéro juste avant l'évaluation, la force de l'individu augmente donc de 0 avant l'essai à $\frac{1}{4}$. Au troisième essai, le signal 10 est émis par l'environnement. Là encore, des deux classeurs activables, à savoir les numéros 3 et 4, le plus spécifique, c'est à dire le numéro 3, va être choisi pour répondre. Le signal émis est 0, il va donner une récompense de $\frac{1}{4}$ à l'individu, lui faisant passer sa force de $\frac{1}{4}$ à $\frac{1}{2}$. Enfin au quatrième et dernier essai, l'individu lit le signal 11 provenant de l'environnement. Ce signal permet d'activer les classeurs 2 et 4. La spécificité seule ne permet pas de déterminer quel classeur activer. C'est donc le hasard qui s'en charge et qui choisit le classeur 2 et émet 0. Le signal émis en réponse par le classeur 2 reste lettre-morte, il n'y a pas de récompense cette fois-ci. La force de fitness finale est de $\frac{1}{2}$.

Cet exemple nous permet de mieux comprendre l'algorithme 2 de fonctionnement des systèmes de classeurs de type Pittsburgh. Les individus sont évalués en pseudo-parallèle. C'est la façon dont l'environnement est ré-initialisé à chaque début d'une phase de tests qui simule le parallélisme. La population des individus évolue lorsque tous les individus ont été évalués un nombre significatif k de fois.

Lors de l'évaluation en système multi-agents, les individus de chaque agent ayant le même rang sont évalués en même temps. L'opération se poursuit jusqu'à ce que tous les individus de chacun des agents ait été évalué k fois.

L'algorithme génétique est appliqué à la fin d'une série d'évaluation de l'ensemble de la population. L'opérateur de croisement peut être légèrement adapté pour permettre différents types de brassage. En effet, lorsque l'on croise des individus, on a le choix soit de préserver, soit de couper les classeurs (cf. § 3.2.3). Dans les deux cas, on peut utiliser un croisement *n-point* ou simple point que nous connaissons déjà (cf. § 3.2.1). L'opérateur de croisement brasse les classeurs afin de permettre l'apparition d'un meilleur individu. La sélection se chargera, quant à elle, d'éliminer progressivement les classeurs puis les individus moins performants. Cette double action explique que les individus soient pratiquement identiques à la fin d'une expérimentation.

Le nombre de classeurs ou règles de production que contient chaque individu semble crucial dans la détermination de la capacité à réagir des individus. A priori, un nombre de règles légèrement trop grand peut permettre en environnement changeant, de disposer d'un capital adaptatif supplémentaire.

Algorithme 2 du Pittsburgh simplifié.**Début**Remplir(P); { *Initialise aléatoirement la population initiale P .* } $Generation = 1$;**Répéter****Pour** (Tous les I_k de P) **Faire**Réinitialiser_Environnement(k); { *Remet l'environnement dans l'état où l'individu I_k l'a laissé ou à zéro.* } $Sanction = 0$;**Pour** (Un certain Nombre de Tests NbT) **Faire**Remplir(Mesg-List); { *avec le message provenant de l'environnement.* }Remplir(Match-List); { *Crée un Match-Set des classeurs de I_k qui peuvent répondre au signal.* }

}

Si (EstVide(Match-List)) **Alors** $Sanction = Sanction + 0$;**Sinon****Si** (Taille(Match-List) > 1) **Alors** $C = ChoisirMeilleur(Match-List)$; { *Choisit le classeur dans Match-List* }**Sinon** $C = Premier(Match - List)$;**FinSi** $Sanction = Sanction + RécompenseAction(C)/NbT$; { *RécompenseAction agit sur l'environnement avec l'action de C et renvoie la sanction liée à cette action.* }**FinSi****FinPour**ChangerForce($I_k, Sanction$); { *Change la force de l'individu I_k en Sanction.* }**FinPour**AppliquerAG(P); $Generation = Generation + 1$;**Jusqu'à** (Critères d'arrêt satisfait);**Fin**

En effet, cette particularité devrait permettre à des règles inutilisées à un instant t d'être toujours présentes à un instant $t + 1$. Par extrapolation, des règles qui ont été utiles en $t - 1$ devraient être disponibles au temps $t + 1$. Nous parlerons alors de mémoire du système de classeurs (cf. § 4.3.2). Nous allons voir maintenant un exemple d'application des systèmes de classeurs de type Pittsburgh simplifié.

4.3.4 Paramètres

Comme nous l'avons déjà souligné, nous nous sommes fixés pour but d'étudier un agent autonome afin d'en dégager les caractéristiques principales à l'aide de mesures. C'est pour cette raison que les réglages de l'algorithme génétique qui se trouvent dans la table 4.3 sont proches de ceux que l'on peut rencontrer dans la littérature. Il est à noter que le nombre de générations est égal à 100 fois le nombre

de générations entre deux changements d'équilibre afin de permettre au système de rencontrer 25 fois un des 4 équilibres. L'élitisme consiste à garder les éléments considérés comme les plus performants, mais nous reviendrons plus en détails sur l'élitisme dans le chapitre 5.

Paramètres	Réglages
$P_{\#}$	33%
$P_{\text{Croisement}}$	70%
P_{Mutation}	0,5%
Mécanisme de Sélection	Roulette Wheel
Sélection des règles (cf. § 4.3.3)	Aléatoire
Croisement	Monopoint
Élitisme	Garde 2 individus
Nombre de Générations (CE voir table 4.5)	$100 \times CE$

TAB. 4.3 – Paramètres de l'algorithme génétique.

Le réglage des paramètres pour les agents est quant à lui, dépendant du problème que nous étudions. La table 4.5 résume les choix que nous avons fait. Le nombre d'essais entre générations est dépendant du nombre de règles afin de permettre de tester potentiellement la majorité des règles de l'individu en cours d'évaluation quel que soit le nombre de règles. Chaque essai correspond à une semaine qui s'écoule.

Paramètres	Réglages
Nombre d'agents	127
Nombre d'individus ($NombreIndividus$)	6,12,18
Nombre de règles par individu ($NombreRegles$)	6,12,18
Nombre d'essais entre générations	$\frac{NombreRegles}{2}$
Période en générations entre deux changements d'équilibre (CE)	50,100,150,250

TAB. 4.5 – Paramètre du système multi-agents.

Nous avons effectué 36 séries différentes de 30 exécutions qui représentent l'ensemble des combinaisons $\{NombreIndividus, NombreRegles, CE\}$. Nous avons moyenné toutes nos expériences sur 30 exécutions. Les résultats que nous présentons dans la sous-section suivante sont un extrait des données les plus remarquables.

4.3.5 Résultats

Nous allons dans un premier temps regarder les résultats obtenus pour la résolution du problème dans un cadre idéal :

- Lorsque le nombre d’individus et de règles vaut 18.
- Le nombre de générations entre deux changements d’équilibre est fixé à 250.

Les figures 4.3 et 4.4 représentent le nombre d’agents qui vont effectivement au bar au cours du temps.

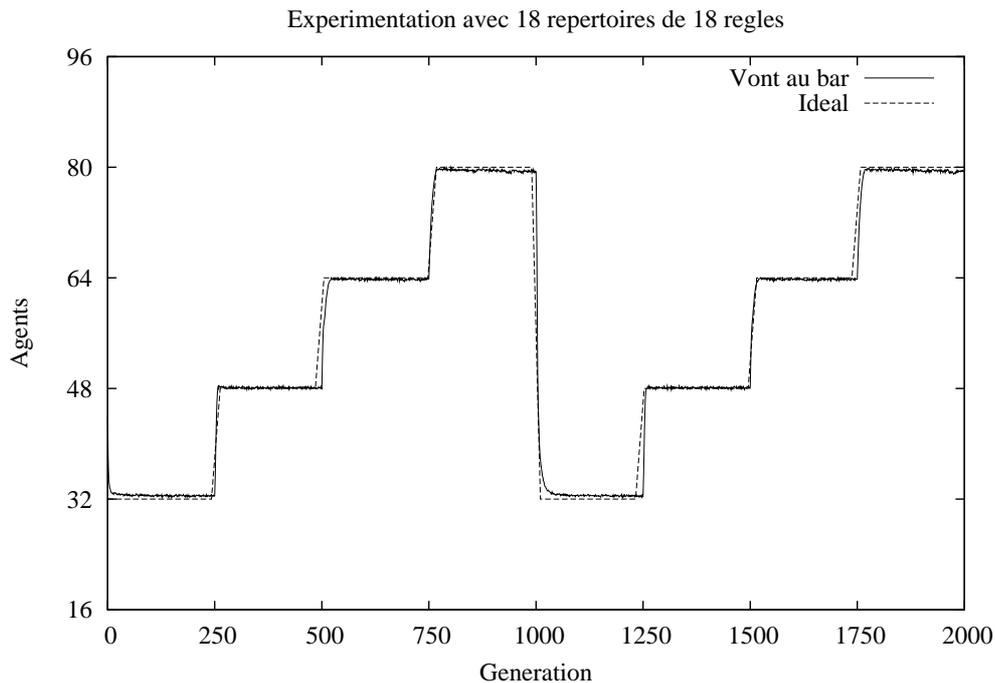


FIG. 4.3 – Résolution du problème du bar, premières générations.

On constate que le système multi-agents se coordonne de manière précise quelque soit l’équilibre idéal à atteindre au cours du temps, et ce, dès les premières générations. Le temps laissé entre deux changements d’équilibre semble largement suffisant pour que le système se stabilise. Lorsqu’on regarde de plus près les phases de transitions, on s’aperçoit que la transition de l’équilibre de 32 agents à 48 ne demande que 9 générations (voir fig. 4.5). De même, le passage de l’équilibre 48 à 64 se fait en moins de 20 générations (voir fig. 4.6). La transition de 64 agents à 80 est effectuée elle aussi en moins de 20 générations (voir fig. 4.7). Enfin, la transition qui semble la plus difficile, de 80 agents à 32 se passe en à peine 24 générations comme le montre la figure 4.8. La composition de chaque agent est donc suffisante pour répondre au problème de coordination à partir de la seule affluence au bar la semaine précédente. La fréquence de changement est suffisamment faible pour permettre au système de se stabiliser et le nombre d’individus et de règles par individu permet également au système de s’adapter aisément aux changements d’équilibre.

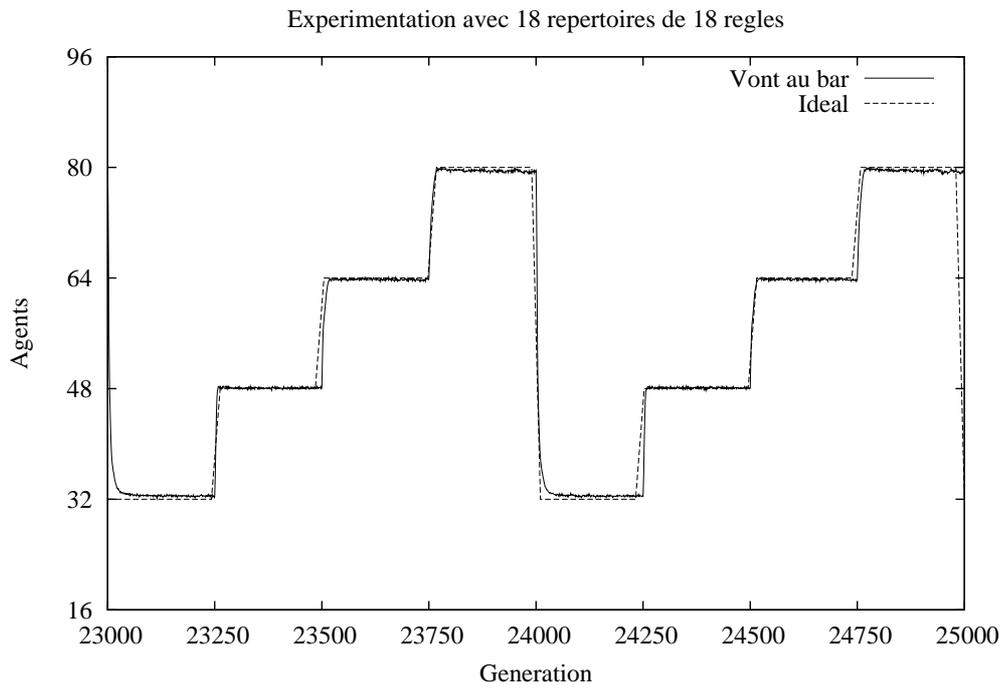


FIG. 4.4 – Résolution du problème du bar, dernières générations.

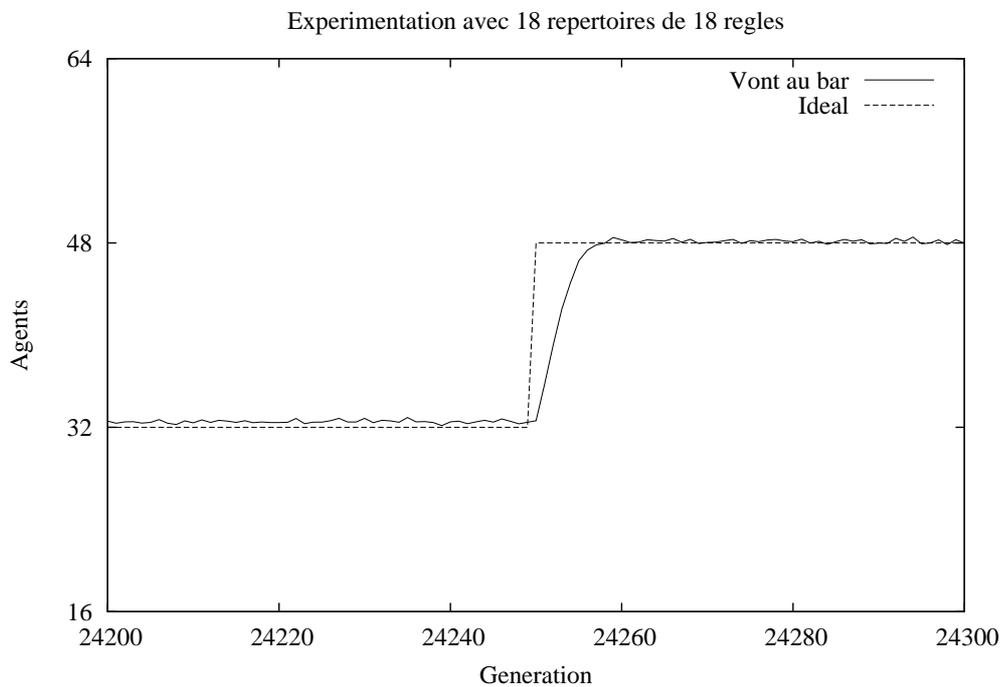


FIG. 4.5 – Transition de 32 agents à 48.

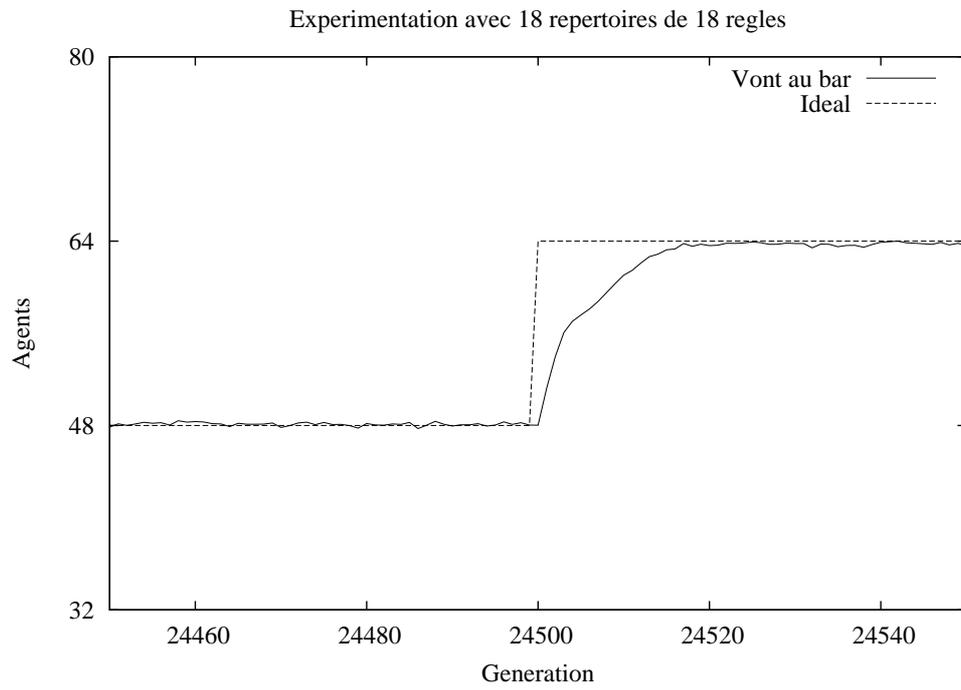


FIG. 4.6 – Transition de 48 agents à 64.

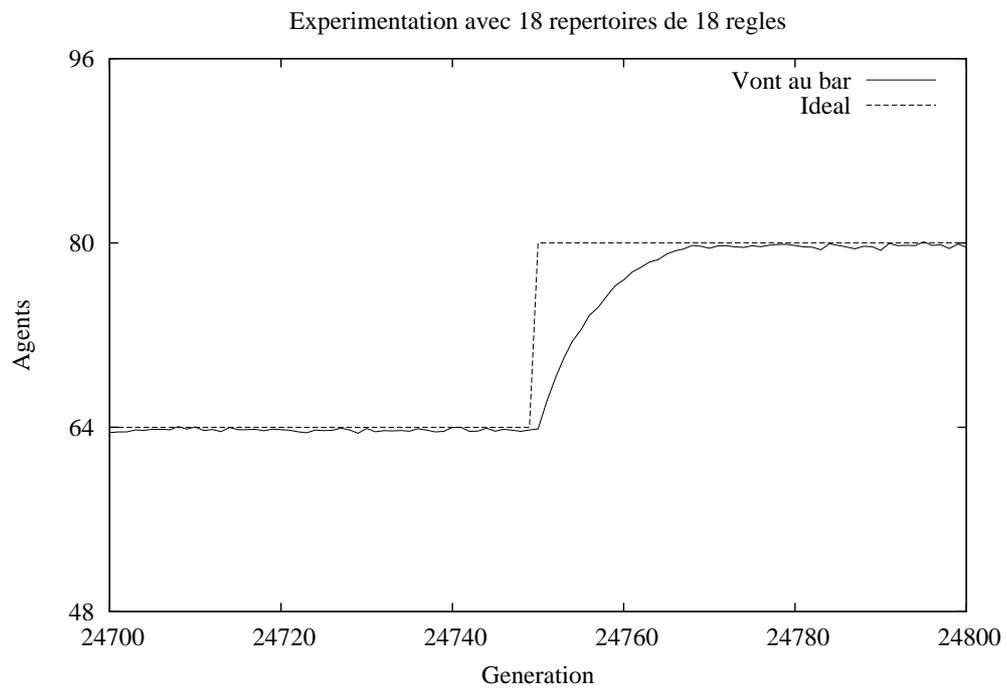


FIG. 4.7 – Transition de 64 agents à 80.

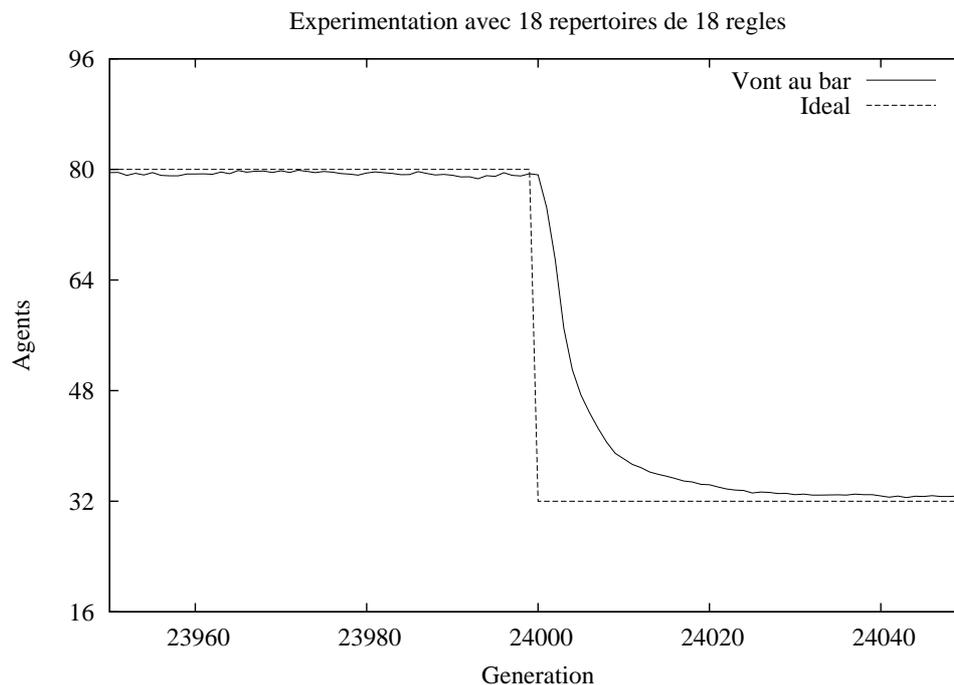


FIG. 4.8 – Transition de 80 agents à 32.

La figure 4.9 présente le comportement moyen des agents en fonction de l'équilibre à atteindre. L'agent 1 est un excellent représentant de la majorité de la population. Lors de la phase où l'équilibre est 32, l'agent 1 va 84% du temps au bar et 16% du temps décide de ne pas y aller pour un total de 100%. Lors des phases suivantes avec un équilibre de 48, 64 et 80, son comportement moyen de venue au bar évolue respectivement à 64%, 32% et 12%. Ce comportement semble correspondre à celui observé chez la majorité des autres agents (voir fig. 4.3). Toutefois les agents semblent capables de ne pas avoir une position tranchée, une stratégie, pour chacune des valeurs d'équilibre à atteindre. Il s'agit bien d'un comportement non-déterministe qui permet à chaque agent de faire partie à tour de rôle, pour une même phase d'équilibre, des agents qui vont ou ne vont pas au bar. Pourtant, il y a bien équilibre :

Perhaps the most exciting implication is the possibility that life had its origin in the vicinity of a phase transition, and that evolution reflects the process by which life has gained control over a successively greater number of environmental parameters affecting its ability to maintain itself at a critical balance point between order and chaos [Langton, 1990].⁴

On peut penser que les agents subissent l'influence de l'environnement dans lequel ils évoluent et que par conséquent aucune individualité ne peut émerger du fait par exemple de la fonction de fitness et du système d'activation des règles. Nous avons pourtant trouvé quelques agents atypiques.

⁴Peut-être que l'implication la plus intéressante est la possibilité que la vie tienne ses origines proches d'une transition de phase, et que l'évolution reflète le mécanisme par lequel la vie a pris le contrôle d'un nombre croissant de paramètres environnementaux qui ont affecté sa capacité à se maintenir au point critique d'équilibre entre l'ordre et le chaos.

L’agent 16 a cette particularité d’aller 100% du temps au bar lorsque l’équilibre à atteindre est 32. Il fait partie de ces rares agents à avoir une stratégie pour un équilibre donné. Cette tendance à aller au bar se retrouve à l’équilibre 48, 80% du temps. Pourtant cette différence s’efface pour les deux derniers équilibres pour rejoindre le comportement moyen de la majorité de la population. L’agent 70 a ceci de particulier qu’il préfère ne pas aller au bar mais seulement pour les équilibres 64 et 80 pour respectivement 52% et 56% du temps.

Cette simple observation nous permet de comprendre l’utilité de mesurer l’évolution de la part de non-déterminisme dans chacun des agents. Nous avons aussi voulu mesurer l’influence sur leur comportement de l’uniformisation des individus qui composent chaque agent.

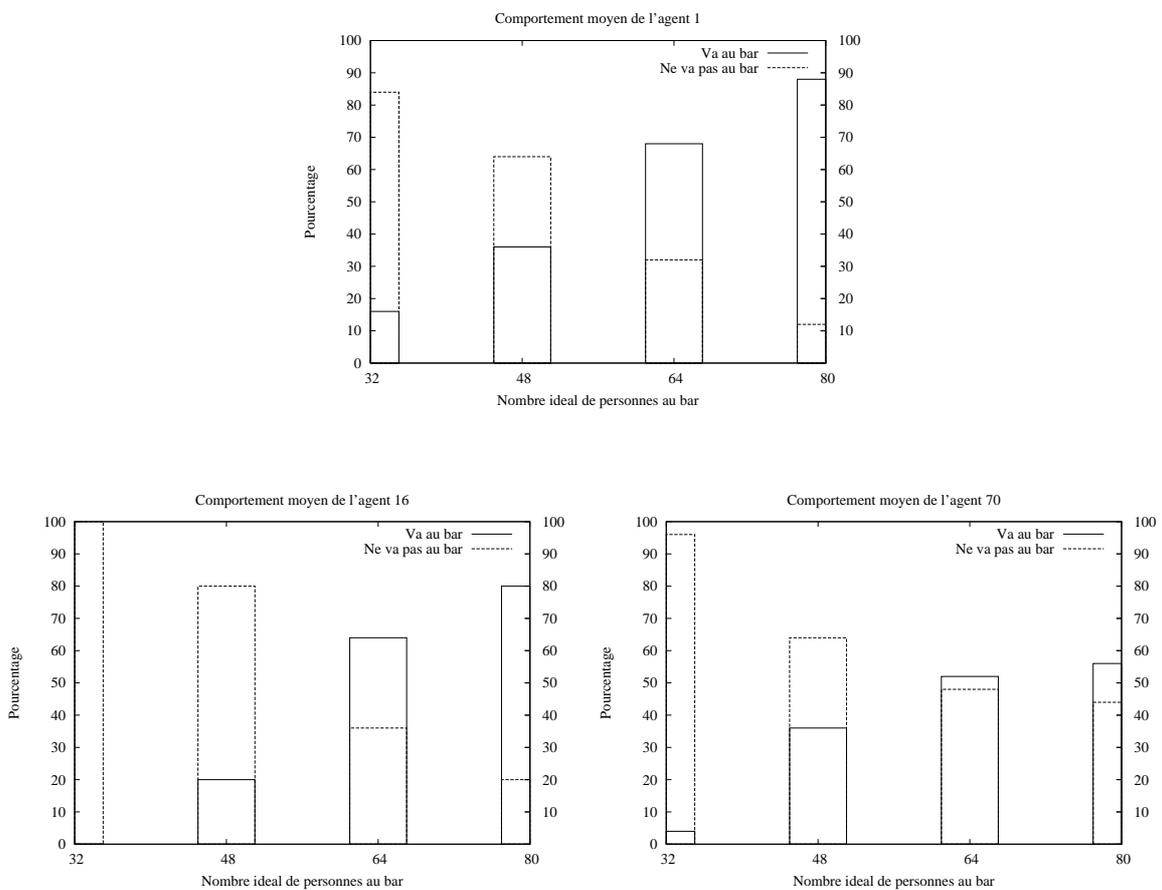


FIG. 4.9 – Exemples de comportement des agents.

Nous allons vérifier la difficulté du problème pour nos agents, en modifiant le nombre d’équilibres possibles dans un cycle de changements d’équilibre.

Influence du nombre d’équilibres Nous avons voulu mesurer les limites de la coordination en faisant varier le nombre d’équilibres possibles dans l’intervalle $[32, 80[$ avant de revenir à l’équilibre 32. Nous avons gardé les mêmes paramètres que lors de l’expérimentation précédente. Le nombre

d'équilibres dans un cycle sur l'intervalle $[32, 80[$ a été fixé à 2, 4, 8, 16 et 24. Les résultats présentés dans la figure 4.10 montrent l'évolution de la taille de la population d'agents (axe des ordonnées) allant au bar à chaque changement d'équilibre (axe des abscisses). Chaque courbe représente soit le nombre d'agents allant au bar, soit le nombre idéal d'agents attendus pour le nombre d'équilibres indiqué entre parenthèses. La formule pour calculer le nombre idéal d'agents devant aller au bar lors de chaque changement d'équilibre (CE) pour un nombre d'équilibres $NbEq$ est :

$$32 + (CE \bmod NbEq) \times \frac{48}{NbEq}$$

De tous les résultats présentés, seul celui des 24 équilibres montre un écart avec le nombre idéal d'agents attendus au bar. Même s'il indique la limite supérieure d'adaptation des agents au problème du bar, cette écart n'est pas significatif.

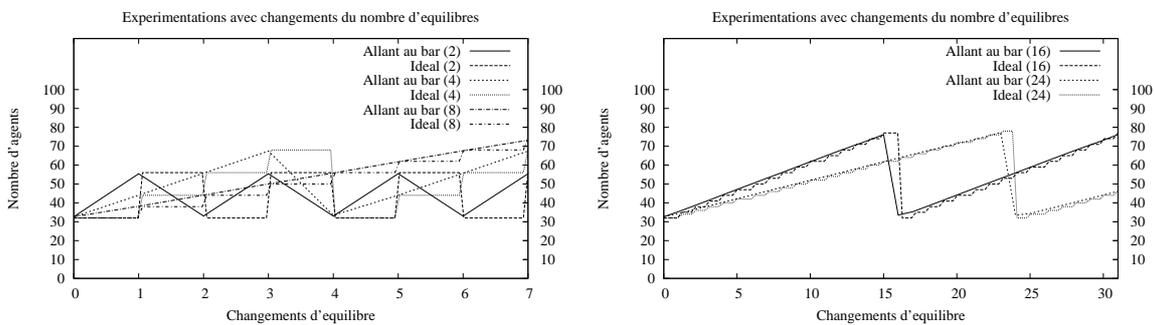


FIG. 4.10 – Influence du nombre de changements d'équilibre.

Le nombre de changements d'équilibre n'est pas la seule cause d'inadaptation des agents au problème du bar. Toutefois ce nombre augmente progressivement la difficulté du problème et on constate que lorsque le nombre d'équilibres dépasse le nombre de règles qui compose chaque individu, le système semble légèrement moins adapté. Voyons ce qu'il en est lorsqu'on change la fréquence des changements d'équilibre avec un nombre de changements d'équilibre fixé.

Influence de la fréquence de changements d'équilibre Cette première étude nous a permis de nous demander quels étaient, a priori, les paramètres qui pouvaient influencer la performance du système. Nous avons décidé, dans un premier temps, de partir d'un agent le plus simple possible et d'étudier l'influence du changement d'équilibre à atteindre sur celui-ci. L'agent est un système de classeurs de type Pittsburgh simplifié, composé de 6 individus contenant 6 règles. Les changements d'équilibre ont lieu successivement toutes les 50, 100, 150 et 250 générations. Les résultats sont présentés sous la forme de moyenne de comportement de coordination sur les 4 derniers changements d'équilibre à savoir les 4 différents équilibres possibles. L'axe des ordonnées représente le nombre d'agents et l'axe des abscisses, le nombre de générations entre deux changements d'équilibre. Chaque figure contient donc 4 courbes qui représentent chacune, le comportement des agents pour les équilibres 32, 48, 64, 80 pour chacune des valeurs de changements d'équilibre.

La figure 4.11 montre qu'il est important de laisser le temps au système multi-agents de s'adapter

au problème qui lui est posé. En effet, lorsqu’il y a changement d’équilibre toutes les 50 générations, on voit très nettement l’incapacité du système à répondre au problème avec précision. Les écarts à l’équilibre vont d’environ 2,5 agents pour l’équilibre 48, à 16 agents pour l’équilibre 80. Le passage à un changement d’équilibre toutes les 100 générations permet déjà au système de mieux se coordonner. En effet, les écarts à l’équilibre, vont cette fois d’environ 2 agents pour l’équilibre 48 à seulement 9 agents pour l’équilibre 80. Le gain est très significatif en terme de précision puisque l’on gagne globalement 39,42%. Comme la figure 4.11 le montre, le gain est un peu moins significatif, bien qu’important lorsque le changement d’équilibre intervient toutes les 150 puis 250 générations. Cette évolution permet d’avoir au final un écart à l’équilibre allant d’un peu moins de 1 agent pour l’équilibre 48 à un peu plus de 4,5 agents pour l’équilibre 80 soit un gain total de 69,29% sur l’ensemble des équilibres.

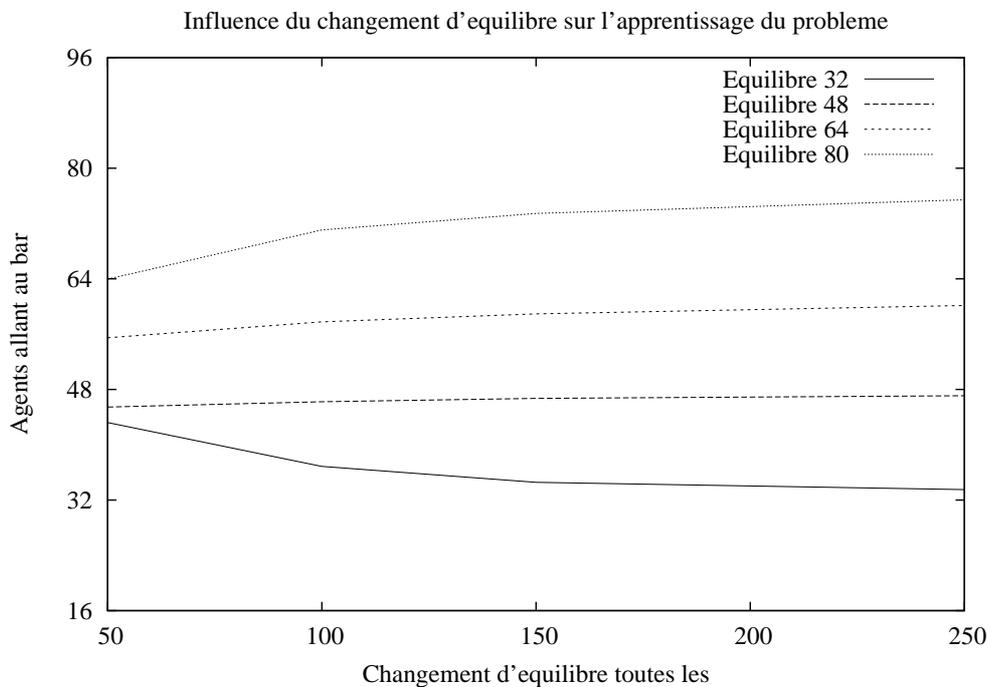


FIG. 4.11 – Changements d’équilibre : Agents allant au bar.

Regardons maintenant si l’écart entre deux changements d’équilibre à la même influence sur les agents décidant de rester chez eux. Le nombre d’agents qui devrait théoriquement rester chez eux est de :

$$NbAgents_{ChezEux} = NbAgents - Equilibre$$

Ce qui nous donne pour les équilibres 32, 48, 64 et 80 respectivement les valeurs 95, 79, 63, 47 pour le nombre d’agents restant chez eux puisque nous avons 127 agents essayant d’aller au bar. Nous utiliserons ces valeurs pour le descriptif suivant. La figure 4.12 nous montre l’évolution du nombre d’agents restant chez eux. Pour CE valant 50, les écarts au nombre idéal d’agents restant chez eux vont

de 32,38 agents pour l'équilibre à 47 qui devraient rester chez eux à 61,79 agents pour l'équilibre à 95. Le meilleur résultat est obtenu lors de l'équilibre 47 alors que le meilleur résultat pour les agents allant au bar était obtenu avec l'équilibre 95. Le passage de CE à 100 permet une évolution importante, car l'écart au nombre idéal d'agents passe de 34,91 agents à 51,55 agents pour les équilibres 47 et 95. Pourtant ce nouveau temps disponible pour apprendre à se coordonner ne favorise pas l'équilibre 47. Si l'on regarde l'ensemble des 4 équilibres, le gain n'est que de 2,58%. Pour les CE 150 et 250 les courbes évoluent dans le bon sens pour atteindre au final un écart au nombre idéal d'agents restant chez eux allant de 30,44 pour l'équilibre 47 à 41,4 agents pour l'équilibre 97. L'évolution est importante puisque l'augmentation de CE a permis de diminuer d'un tiers le plus grand écart au nombre idéal d'agents restant chez eux. Le gain sur l'ensemble des équilibres pour le passage de CE de 50 à 250 représente 15,75%. Il faut se demander toutefois ce que sont devenus ces 41,4 agents qui auraient dû rester chez eux et les 9 autres qui auraient dû aller au bar. Cette nouvelle catégorie d'agent représente une forme de nouveau choix d'action : le choix de ne rien faire.

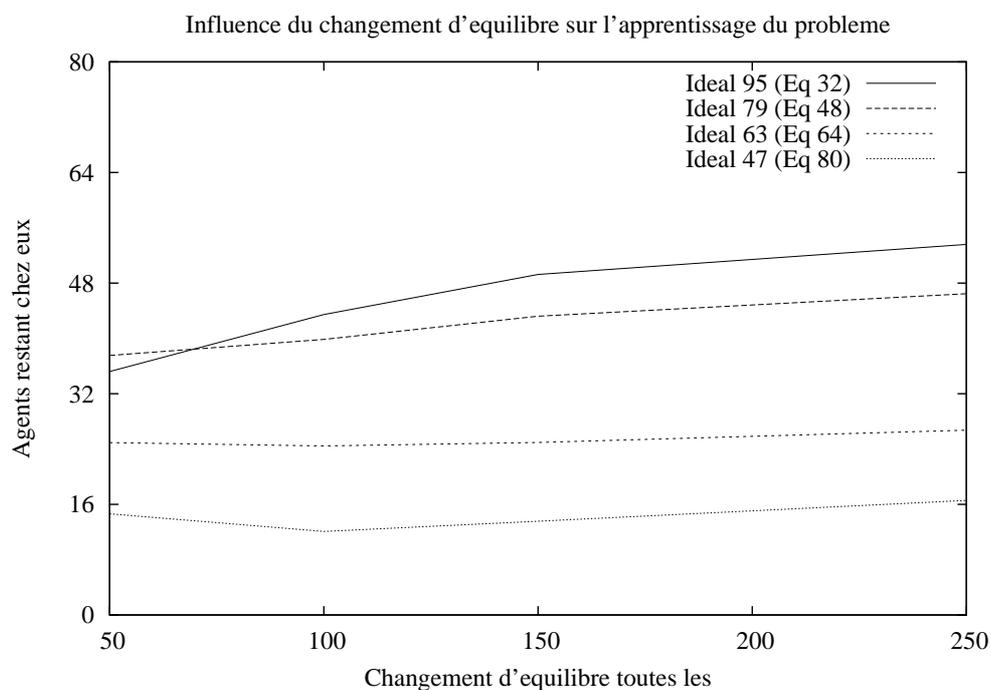


FIG. 4.12 – Changements d'équilibre : Agents restant chez eux.

Cette nouvelle possibilité d'action des agents s'exprime à travers la figure 4.13 et nous appellerons ces agents, des agents *indécis*. Les agents indécis regroupent les agents qui auraient dû aller au bar et qui n'ont pas sû y aller et d'autre part les agents qui auraient dû rester chez eux et qui sont restés chez eux mais n'ayant pas pris cette décision, ils n'ont pas eu de récompense. Le nombre d'agents indécis suit la même courbe décroissante quelle que soit la valeur d'équilibre tandis que CE augmente. On voit donc le nombre moyen d'agents indécis successivement passer de 46,9 à 44,05 puis 40,84 et

enfin 37,12 pour des valeurs de CE respectivement de 50, 100, 150 et 250, soit une diminution de 20,87% du nombre d’agents indécis. Le temps est donc un facteur permettant au système de mieux se coordonner autour des objectifs à atteindre. Toutefois, le nombre d’agents indécis reste trop élevé pour parler de coordination précise du système. L’agent, composé lors de ces expérimentations de 6 individus contenant 6 règles, semble insuffisamment adapté au problème.

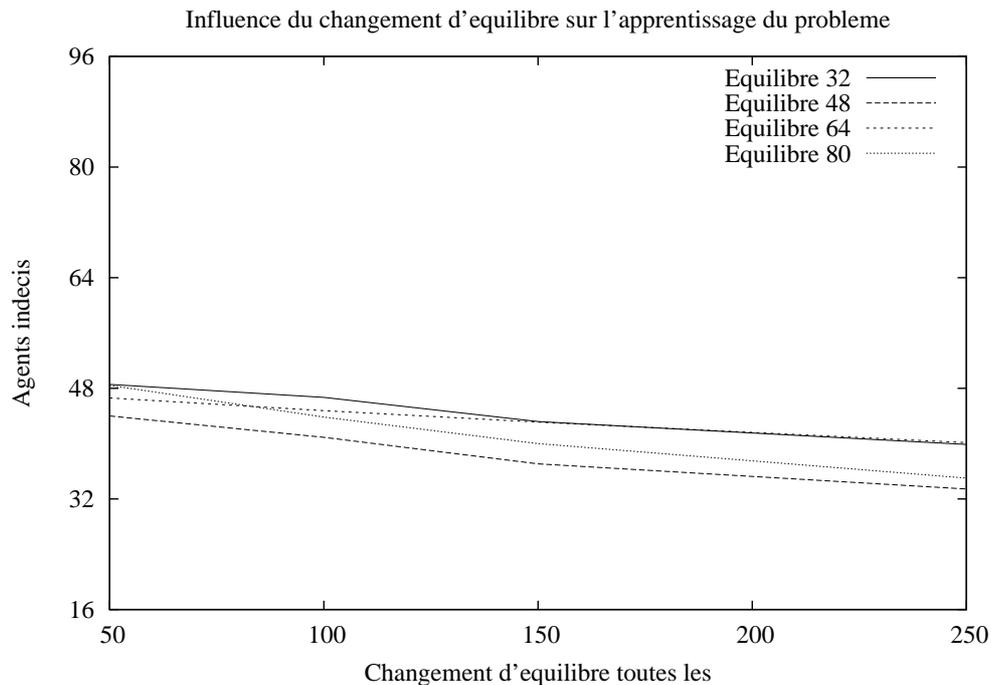


FIG. 4.13 – Changements d’équilibre : Agents indécis.

La figure 4.14 présente l’influence du changement de CE sur la mémorisation des agents. La mémorisation passe ainsi de 50,58% à 45,10% puis 44,14% et enfin 44,43% lors des différents CE . Ce résultat contre-intuitif semble provenir du faible potentiel de l’agent en terme de nombre d’individus et de nombre de règles par individu. En effet, plus on laisse de temps aux agents pour se rendre compte qu’ils se trompent, plus ils essayent de corriger leur erreur à chaque changement d’équilibre et du fait de leur faible capacité, ils sont obligés de réadapter leur capital de règles aux nouvelles conditions rencontrées. Il faut toutefois noter que la tendance générale de baisse s’inverse légèrement lorsque CE passe de 150 à 250.

Pourtant d’après nos résultats, près de $\frac{1}{3}$ des agents n’adopte pas de stratégie malgré une uniformisation des individus de chaque agent que nous avons mesuré à près de 94% dans l’ensemble des expérimentations présentée dans ce chapitre. Nous avons voulu savoir pourquoi, en mesurant le potentiel non-déterministe des agents. La figure 4.15 montre l’évolution de la part de choix non-déterministes disponibles pour les agents lorsque CE évolue. La courbe révèle que le temps n’augmente en aucune façon la part de non-déterminisme des agents même si elle existe.

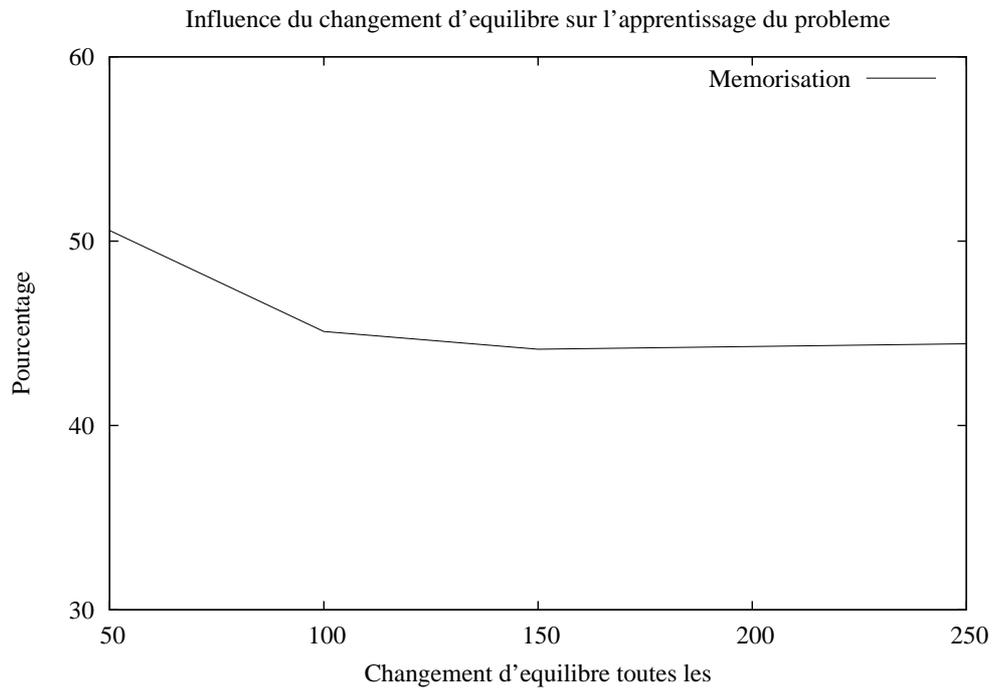


FIG. 4.14 – Changements d'équilibre : Mémorisation.

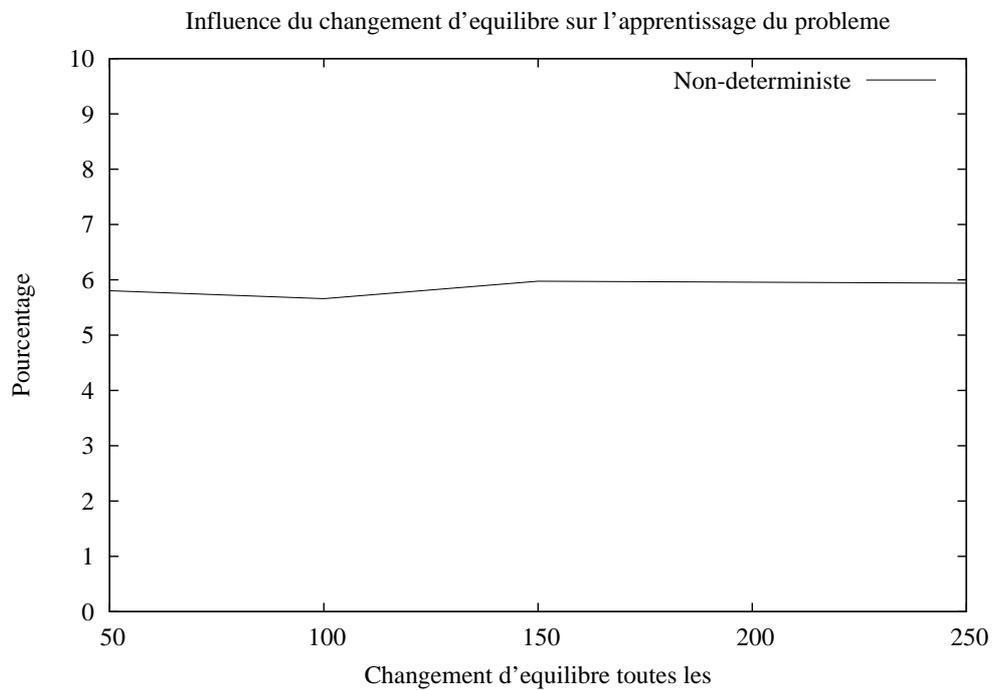


FIG. 4.15 – Changements d'équilibre : Part non-déterministe.

En conclusion, l’augmentation de CE permet :

- d’améliorer la coordination.
- de diminuer le nombre d’agents indécis.
- n’améliore pas la capacité de mémorisation.
- n’influence pas la part non-déterministe des agents.

Voyons maintenant en quoi la modification des paramètres du nombre d’individus et du nombre de règles influence la coordination du système.

Influence du nombre d’individus qui composent l’agent Pour ces expérimentations, nous avons fait varier le nombre d’individus de 6 à 18 par pas de 6 tout en conservant un nombre de règles de 6. CE est fixé à 250 générations afin que les résultats présentés ne soient essentiellement influencés que par le changement du nombre d’individus. La figure 4.16 montre le nombre d’agents allant au bar en fonction du nombre d’individus par système de classeurs. Le gain le plus significatif concerne l’équilibre 80 qui passe de 75,44 agents allant au bar pour 6 individus par système de classeurs à 78,77 agents avec 18 individus. Ce gain, en terme d’agents vers l’équilibre, représente 8,07% supplémentaire lorsqu’on passe de 12 à 18 individus alors qu’il est de 64,96% pour le changement de 6 à 12 individus. L’augmentation du nombre d’individus permet donc une meilleure réponse au problème mais avec un coût non négligeable en complexité puisqu’on multiplie par 3 celle-ci.

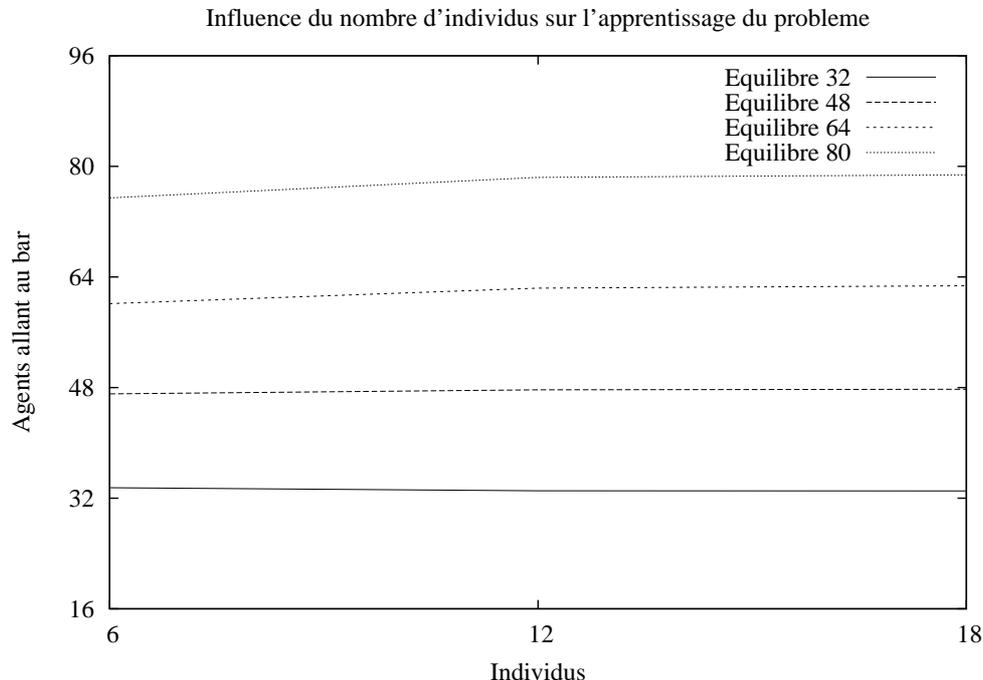


FIG. 4.16 – Variation du nombre d’individus : Agents allant au bar.

On peut alors se demander l’intérêt d’augmenter le nombre d’individus si le coût est plus important

que le gain. La figure 4.17 présente les résultats sur les agents restant chez eux. Les courbes de chaque équilibre suit globalement la même progression. On obtient alors un gain du nombre d'agents restant chez eux de 49,24% en moyenne ce qui représente, par exemple, pour le passage de 6 à 12 individus, une augmentation générale de 14,45 agents vers les différents équilibres. Le gain est de 7,69% pour le passage de 12 à 18 individus soit un gain général de 2,71 agents. Là encore, le gain est plus significatif lorsqu'on passe de 6 à 12 individus que lorsqu'on passe de 12 à 18 individus.

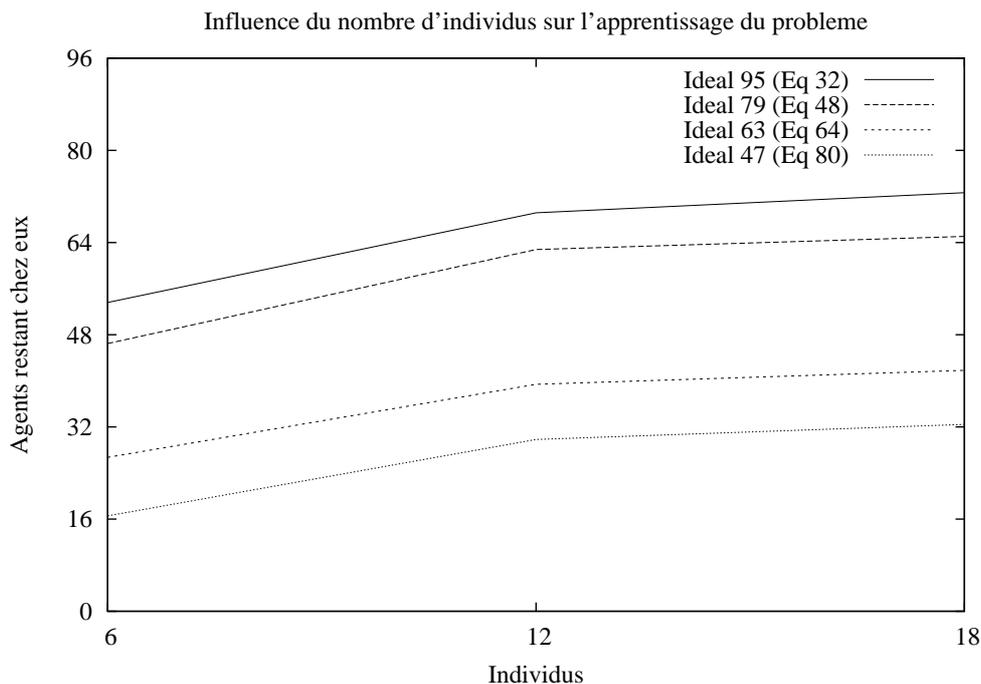


FIG. 4.17 – Variation du nombre d'individus : Agents restant chez eux.

Regardons maintenant si la variation du nombre d'individus a bien l'influence escomptée sur le nombre d'agents indécis. La figure 4.18 nous permet de suivre l'évolution du nombre d'agents indécis. Cette évolution reflète parfaitement le constat fait précédemment sur les agents prenant une décision. Ainsi le passage de 6 à 18 individus divise pratiquement par 2 le nombre d'agents indécis. La diminution la plus importante étant de 42,98% lors du passage de 6 à 12 individus. Le passage de 12 à 18 individus ne permet qu'une diminution du nombre d'agents indécis de 7,80%. Ce dernier résultat doit être considéré comme significatif même si il reste en retrait. Malgré le nombre d'individus relativement élevés, il reste en moyenne 18,43 agents indécis, ce qui représente encore 14,51% des agents. Ce nombre reste trop élevé.

La figure 4.19 montre les effets de l'augmentation du nombre d'individus sur la capacité de mémorisation. La capacité de mémorisation augmente en atteignant le taux de 52,95% au lieu de 44,43% lorsque le nombre d'individus varie de 6 à 12 et ce taux passe à 55,74% pour l'évolution à 18 individus. L'augmentation du nombre d'individus semblent influencer fortement la capacité de mémorisa-

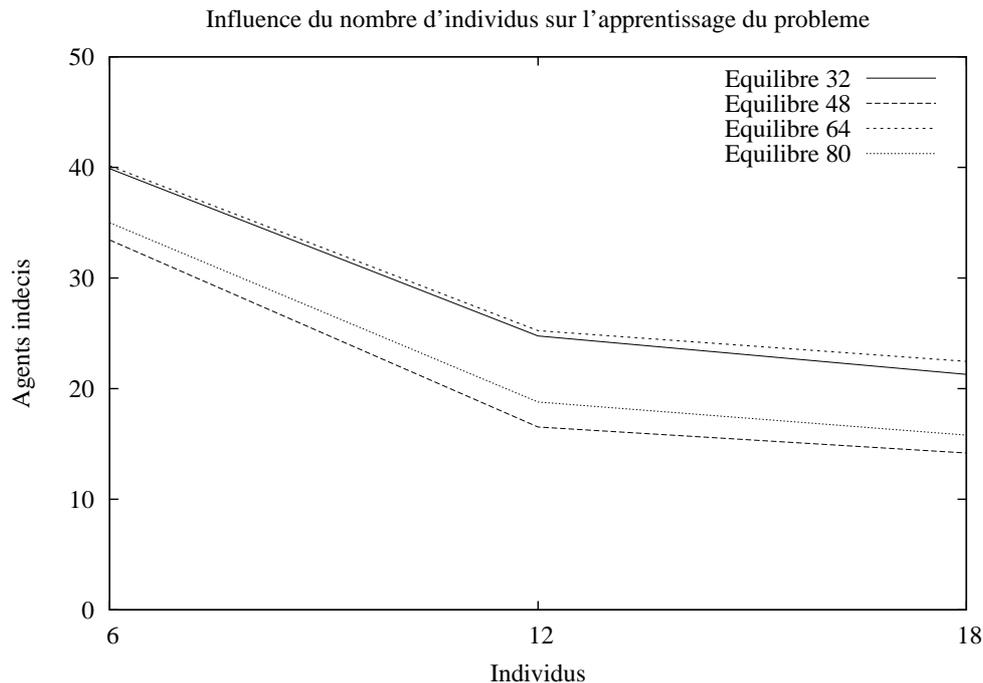


FIG. 4.18 – Variation du nombre d'individus : Agents indécis.

tion du système.

La figure 4.20 montre l'évolution de la part non-déterministe lorsque le nombre d'individus augmente. Avec une augmentation de 1,68 points du pourcentage, on constate que le nombre d'individus influence effectivement la part de choix non-déterministe de chaque agent. Toutefois les 7,62% de choix non-déterministe ne sont pas suffisants pour justifier que plus de $\frac{1}{3}$ des agents ne sachent pas adopter de stratégie. L'augmentation du nombre d'individus permet d'augmenter la part non-déterministe ce qui paraît corrélé à la diminution du nombre d'agents indécis.

Comme bilan, nous pouvons d'ores et déjà établir qu'augmenter le nombre d'individus permet :

- d'améliorer significativement la coordination autour du problème.
- de diminuer de manière importante le nombre d'agents indécis.
- d'augmenter la capacité de mémorisation et de permettre ainsi l'élaboration d'une stratégie pour chaque agent.
- de légèrement augmenter la part de non-déterministe.

L'indécision révèle une incapacité de l'agent à déclencher une règle de comportement, augmenter le nombre d'individus semble pourtant augmenter la capacité de décision des agents.

Influence du nombre de règles qui composent chaque individu Nous allons voir de quelle manière l'évolution du nombre de règles influence la coordination des agents. Pour ces expérimentations, le nombre d'individus est fixé à 6 et *CE* à 250 pour permettre, comme précédemment, de n'étudier

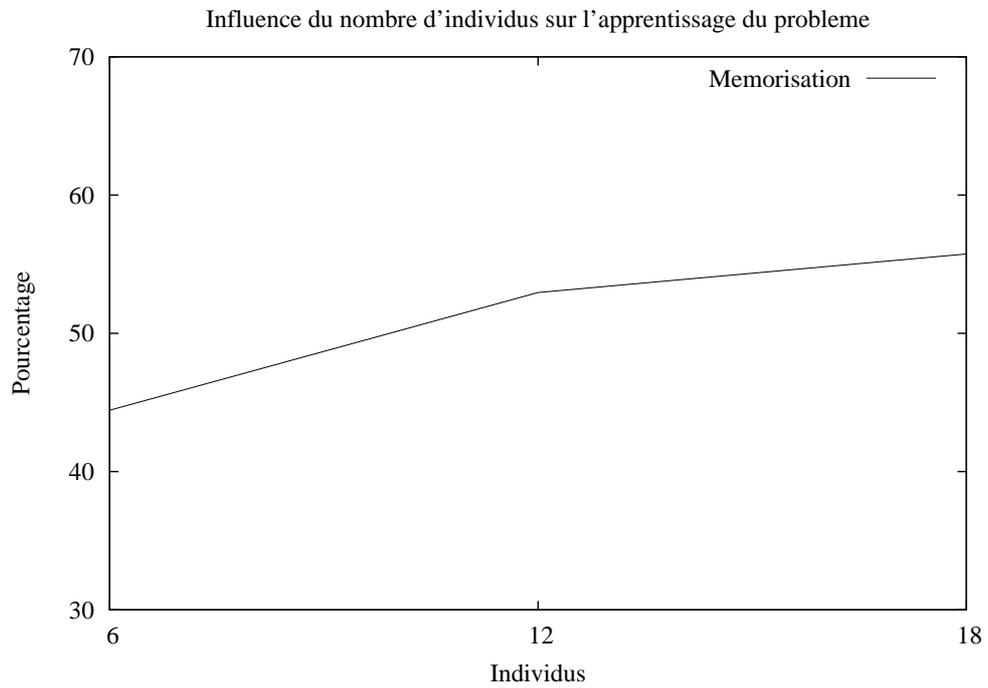


FIG. 4.19 – Variation du nombre d'individus : Mémorisation.

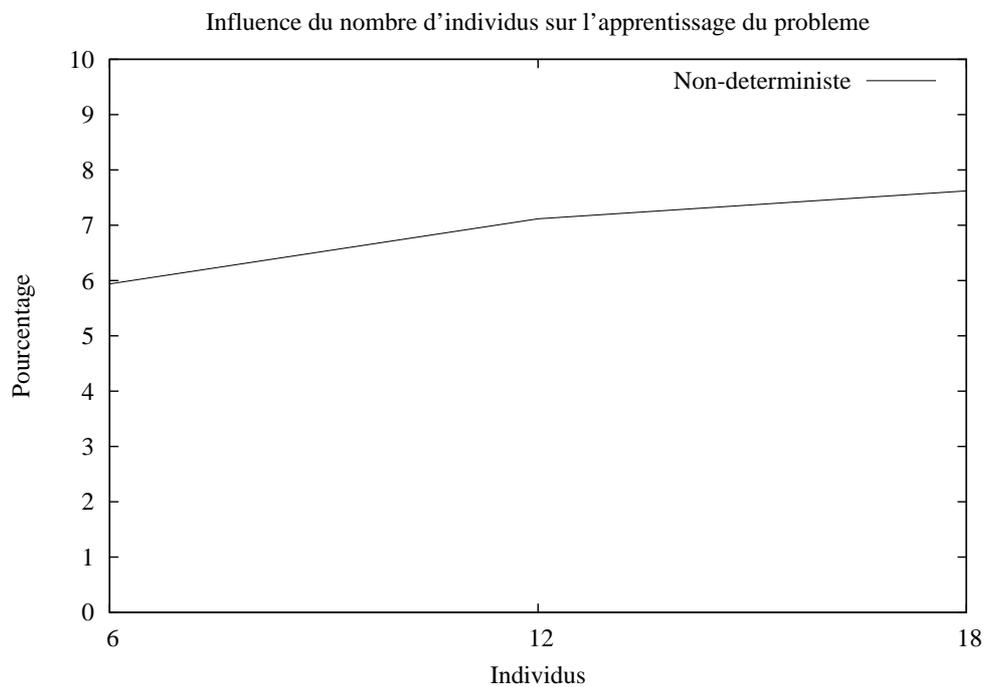


FIG. 4.20 – Variation du nombre d'individus : Part non-déterministe.

que l’influence du nombre de règles sur le système. La figure 4.21 présente l’évolution du nombre d’agents allant au bar lorsque le nombre de règles augmente. Pour l’équilibre 80, par exemple, on passe de 75,44 agents pour 6 règles à 78,14 agents pour 18 règles, ce qui représente un gain de 59,16% moins significatif que lors de l’augmentation du nombre d’individus. Plus éloquent encore, pour l’équilibre 32, le nombre d’agents varie de 33,51 à 34,61, ce qui représente une perte d’efficacité du système de 72,59% !

Globalement, le passage de 6 à 12 règles entraîne un gain en terme d’agents vers la coordination de 29,70% et pour le passage de 12 règles à 18, un gain de 4,62% supplémentaire. On constate donc que l’augmentation du nombre de règles semble permettre à l’agent de prendre plus de décisions mais pas obligatoirement les bonnes.

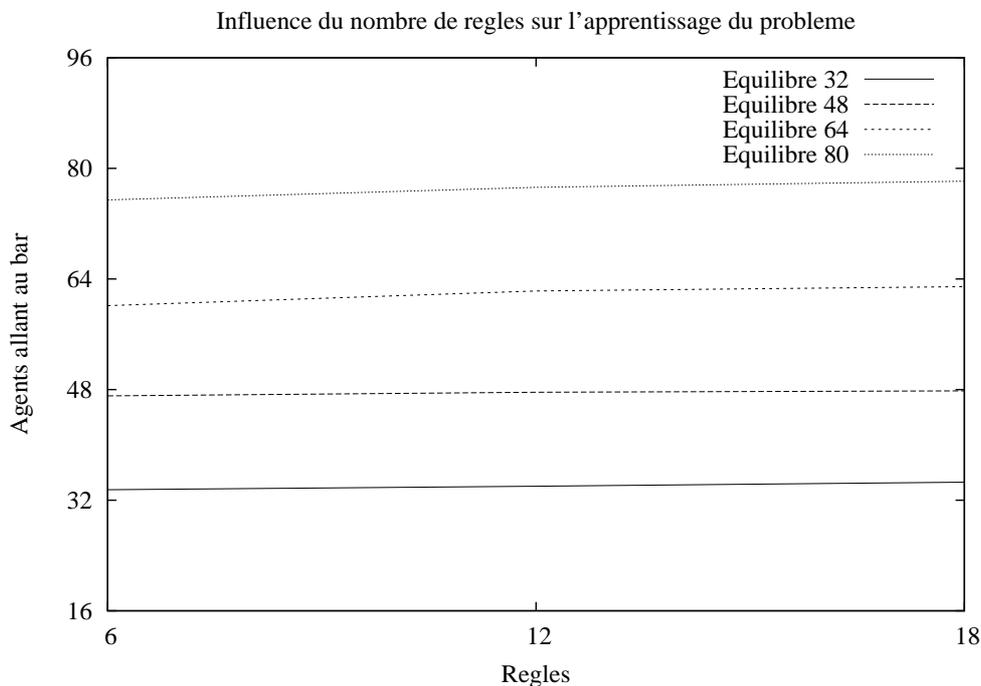


FIG. 4.21 – Variation du nombre de règles : Agents allant au bar.

Nous allons vérifier si ce phénomène se reproduit avec les agents restant chez eux. La figure 4.22 présente l’évolution du nombre d’agents restant chez eux. Les courbes des quatre différents équilibres évoluent de la même manière, ce qui nous permet de ne présenter que les résultats globaux. Le nombre d’agents augmente de manière importante lors du passage de 6 à 12 règles, entraînant un gain de 46,05% vers le nombre d’agents idéal restant chez eux. En allant de 12 à 18 règles, un gain supplémentaire de 19,81% vient s’ajouter au précédent. Le gain total étant de 65,86%, il semblerait qu’effectivement, l’augmentation du nombre de règles ait un rôle important pour permettre aux agents de prendre des décisions mais cette fois-ci, de bonnes décisions.

Regardons maintenant l’influence de ces différentes évolutions du nombre de règles sur la popu-

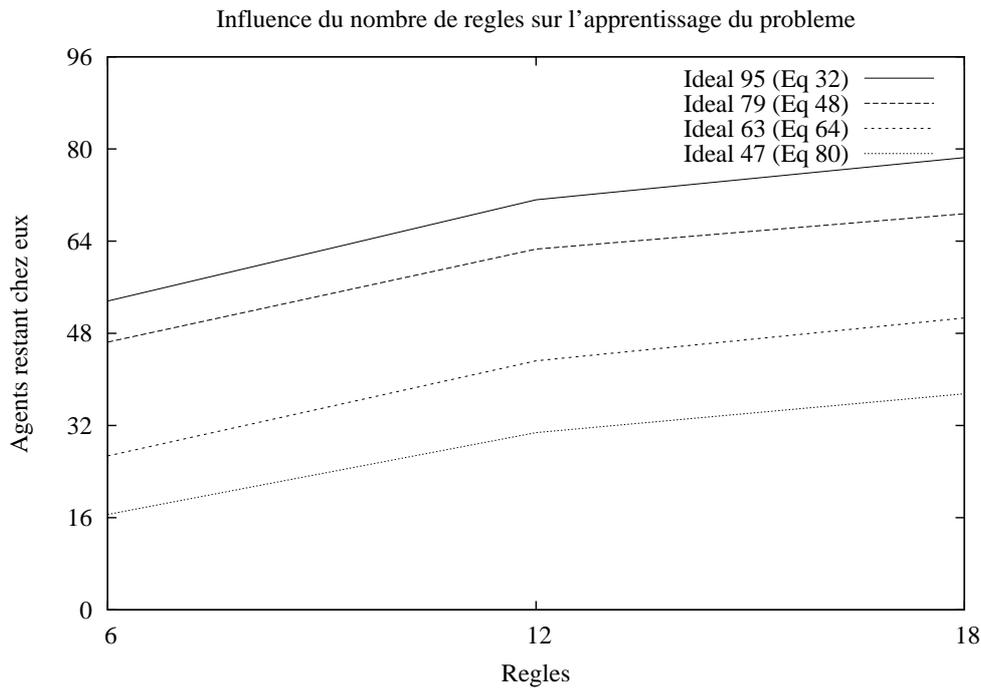


FIG. 4.22 – Variation du nombre de règles : Agents restant chez eux.

lation des agents indécis. La figure 4.23 montre cette évolution. De la même manière que pour les agents restant chez eux, les quatre courbes suivent une pente presque identique. Ainsi, le passage de 6 à 12 règles permet une diminution du nombre d'agents indécis de 46,82% et celui de 12 à 18 indique une baisse supplémentaire de 20,19%. Au final, 12,28 agents en moyenne n'arrive toujours pas à se décider, soit 9,67% de la population. Si l'on compare ce résultat à celui de l'augmentation du nombre d'individus, on constate alors que la taille de la population des agents indécis est divisée par presque 2 lorsque le nombre d'individus augmente de 6 à 18. Tandis que l'augmentation du nombre de règles de 6 à 18 permet de diviser par plus de 3 cette même population.

L'effet de l'évolution du nombre de règles sur la mémorisation est exprimé par la figure 4.24. Le taux de mémorisation du système multi-agents évolue de 44,43% qui est notre valeur de référence à 57,84% puis 62,70%. L'augmentation du nombre de règles influence donc fortement la capacité de mémorisation des agents. Elle permet donc aux agents de développer une stratégie même si certains d'entre eux ne peuvent toujours pas se décider (9,67%).

Enfin la figure 4.25 présente l'évolution de la part de choix non-déterministe des agents lorsque le nombre de règles par individu augmente. L'évolution est sans équivoque et linéaire et progresse légèrement plus vite que le nombre de règles lui-même avec un taux de 23,15% de non-déterminisme avec 18 règles alors qu'il n'était que de 5,94% pour 6 règles par individu. Ainsi, avec 18 règles par individu, l'agent a 23,15% de chance d'être dans une situation ambiguë ou plusieurs règles avec des choix d'action différents vont pouvoir être activées. Pourtant les agents sont capables de se coordonner

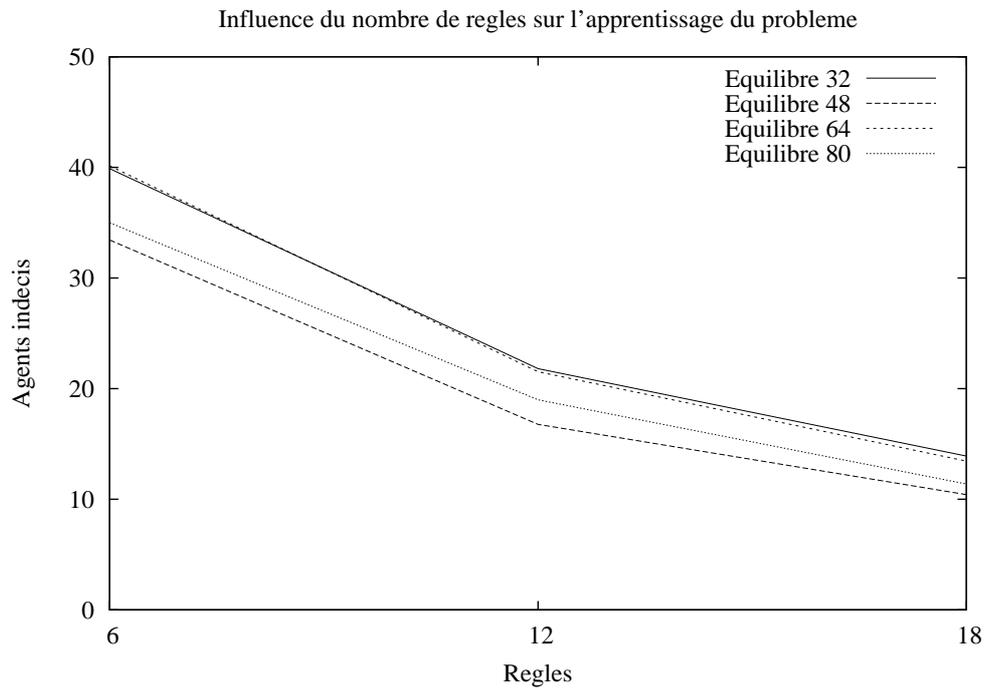


FIG. 4.23 – Variation du nombre de règles : Agents indécis.

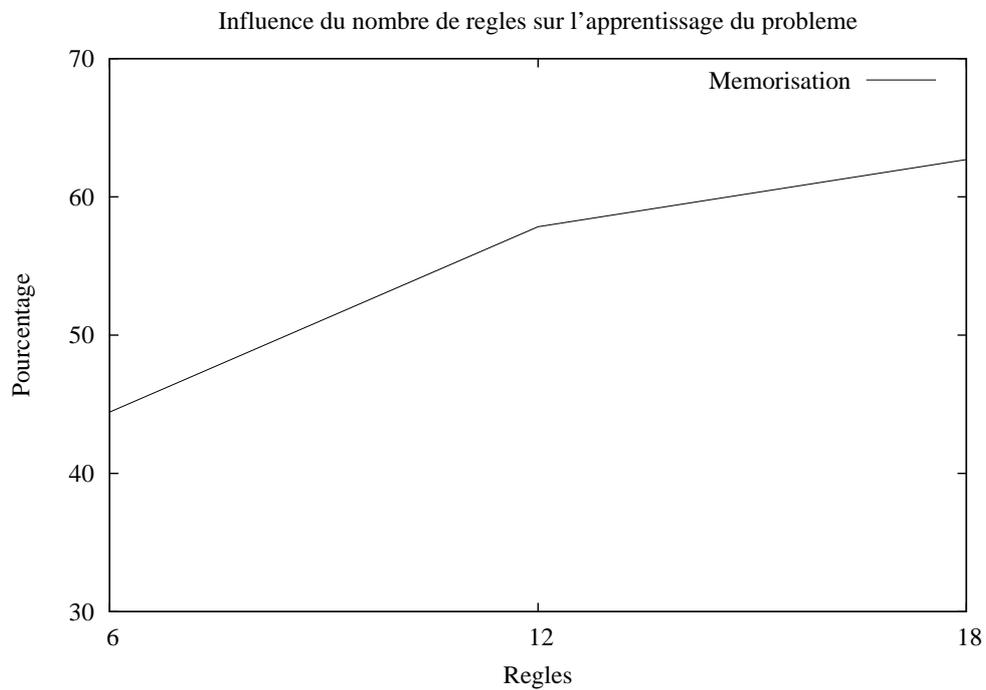


FIG. 4.24 – Variation du nombre de règles : Mémorisation

autour des différents équilibres à atteindre. La méthode aléatoire de sélection lorsque plusieurs règles sont activables, amplifie ce phénomène. On comprend alors mieux pourquoi augmenter le nombre de règles n'est pas le facteur déterminant quant à la résolution du problème de coordination mais qu'il permet d'avoir un plus large éventail de choix lors de la prise de décision. Retenons également que plus le nombre de règles augmente, plus ce nombre devient un facteur de bruit dans la prise de décision et risque d'engendrer un comportement trop ératique du système, empêchant par la même le système de se coordonner de manière précise au sens de [Rustogi et Singh, 1999].

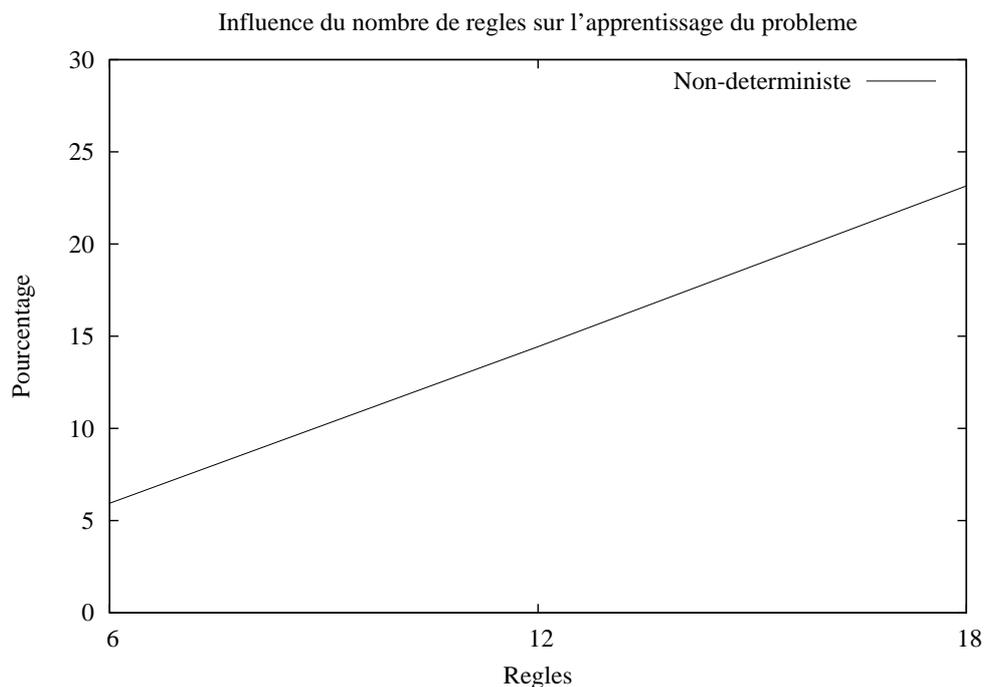


FIG. 4.25 – Variation du nombre de règles : Part non-déterministe.

Ce résultat semble pourtant contredire les résultats observés sur la mémorisation. En effet, l'augmentation du non-déterminisme des agents auraient dû, *a priori*, diminuer la capacité de mémorisation, or c'est exactement l'inverse qui se produit lorsqu'on augmente le nombre de règles. Il semblerait que le phénomène non-déterministe soit limité à des valeurs peu utilisées jusque là par l'agent qui, de lui-même, sélectionne les règles intéressantes grâce à l'algorithme génétique, tout en augmentant son potentiel réactif pour le cas où l'environnement viendrait à tester des valeurs non utilisées jusque là. La méthode de sélection des règles activables a donc une influence sur les règles retenues par le système afin de se coordonner.

L'augmentation du nombre de règles permet :

- d'augmenter fortement le pouvoir de décision des agents.
- d'améliorer la coordination du système multi-agents sans améliorer sa précision.
- d'améliorer fortement la capacité de mémorisation des agents.

- de développer, pour les agents, une stratégie.
- de prévoir l’avenir.

Le temps, le nombre d’individus et le nombre de règles par individus sont des facteurs important pour permettre la coordination du système de manière précise. Ils permettent également de définir des limites à la coordination telles que celles énoncées par [Rustogi et Singh, 1999] dans la section précédente.

C’est grâce à l’uniformisation des individus qui le composent, qu’un agent développe une stratégie de comportement, le nombre de règles et le nombre d’individus lui permettent d’améliorer la précision et de diminuer l’inertie en augmentant l’éventail de choix possibles.

Ces résultats permettent de mieux comprendre le succès quasi parfait lorsque les agents sont des systèmes de classeurs composés de 18 individus et 18 règles par individu et que le temps entre deux changements d’équilibre est de 250. Derrière l’équilibre apparent se cache un comportement individualiste mais rationnel des agents [Forrest et Miller, 1990].

4.4 Conclusion

Nous venons de décrire l’influence des principaux paramètres des systèmes de classeurs de type Pittsburgh simplifié sur le comportement des agents composant un système multi-agents se coordonnant de manière spontanée. Nous avons observé, à l’aide de mesures d’uniformisation, de capacité de mémorisation, de part de non-déterminisme et de réponse au problème posé, comment chaque agent est un individu à part entière avec un comportement qui lui est propre et qui permet au système de se coordonner. Par rapport à un tel problème, l’agent autonome composé d’un classeur de type Pittsburgh simplifié voit principalement sa précision influencée par :

- Le nombre d’individus qui améliore de manière significative la précision lorsqu’il augmente.
- Le nombre de règles qui s’il est insuffisant diminue la précision et s’il est trop grand ne l’améliore plus.
- Le temps imparti pour se coordonner qui est un facteur exogène au système.

L’inertie des agents est déterminée par :

- L’uniformisation de l’agent qui dépend elle-même de l’algorithme génétique et augmente l’inertie.
- La capacité de mémorisation qui plus elle est importante, plus l’agent restera sur ses positions et augmentera l’inertie.
- Le non-déterminisme qui régit également la capacité de l’agent à s’adapter à de nouvelles situations non apprises et qui diminue l’inertie.

Le choix, la connaissance et le partage sont des notions inhérentes au problème abordé. Les paramètres du système de classeurs ne changeront rien à ces variables. Seul un codage adapté ou une capacité de l’agent permettent d’optimiser le système. Comme nous l’avons dit dans le chapitre 2, l’agent durant sa vie peut passer de son rôle génétique, à savoir l’agent autonome spécialisé dans une tâche, à un

agent communicant un savoir ou encore communicant une information. Nous venons d'étudier quels étaient les rôles des principaux paramètres permettant à l'agent d'être autonome. Nous allons voir maintenant comment ce même agent peut échanger un savoir avec ses semblables.

Chapitre 5

Communiquer avec des SMA Homogènes

L'échange du savoir dans un système multi-agents implique la détention, par au moins un agent, d'un savoir utile à la communauté ainsi que la capacité à le transmettre. Cela implique que les autres agents puissent le comprendre. Nous allons voir dans un premier temps ce qu'est un savoir et comment le distribuer puis dans un second temps comment notre agent composé d'un système de classeurs de type Pittsburgh simplifié peut échanger sa connaissance avec d'autres agents.

5.1 Problématique des SMA Homogènes

Lorsqu'on aborde un problème en Intelligence Artificielle Distribuée, le concept de distribution varie selon le point de vue de l'auteur. Par exemple, [Dagaëff et Chantemargue, 1998] nous rappellent que distribuer des agents sur un problème, c'est implicitement les répartir dans le temps et décider arbitrairement s'ils vivent de manière synchrone ou non. Le facteur temps peut permettre à un agent d'avoir une date de "naissance" et une date de "décès" et de choisir le moment où il interagit avec les autres en fonction de sa disponibilité. Cette optique est intéressante lorsque les caractéristiques et les propriétés des agents sont maîtrisées et que l'on souhaite passer à un phase réelle d'exploitation des découvertes effectuées. Lors de nos études, nous avons supposé que les agents "vivaient" de manière synchrone et uniquement durant le temps de l'expérimentation afin d'étudier les agents sur un problème précis à la manière de l'éthologie synthétique.

Notre approche de la distribution a été semblable à celle d'autres auteurs que nous allons évoquer maintenant, à savoir la distribution des agents sur un problème donné. Nous verrons que distribuer des agents sur un problème peut engendrer d'autres répartitions telle que la répartition du savoir qui sera le but de notre étude.

5.2 Quelques exemples

Comme [Maturana, Shen et Norrie, 1999] le soulignent, lorsque des agents sont répartis sur des sous-problèmes et qu'ils échangent des informations, il est nécessaire que ceux-ci soient coordonnés

entre eux sachant que les liaisons qui s'établissent peuvent évoluer au cours du temps. C'est le cas de la fourmi vue au chapitre 2. Les liaisons doivent s'adapter aux besoins locaux des agents qui n'ont pas toujours besoin d'information lorsqu'ils se suffisent à eux même. Une fourmi qui a trouvé une source de nourriture n'a pas intérêt à arrêter de l'exploiter même si elle rencontre une autre piste indiquant qu'il y a de la nourriture ailleurs. Cette information risque de perturber son analyse et laisser de la nourriture inexploitée.

Dans l'étude menée par [Hogg et Huberman, 1993], la question est posée de savoir comment exploiter la connaissance détenue par un agent pour qu'elle profite aux autres agents du système. L'idée étant évidemment, d'exploiter au maximum la coopération entre les agents. A travers deux exemples, les auteurs proposent deux solutions pour le partage de la connaissance. Leur postulat est que l'on peut transmettre des solutions partielles entre les agents afin d'éliminer des branches de recherche inutiles dans la résolution de problèmes P et NP (polynomiale non-déterministe). Ils étudient tout d'abord un problème de cryptarithmétique, problème connu de résolution de contraintes qui consiste à trouver la solution d'addition cryptée avec des lettres :

$$DONALD + GERALD = ROBERT \Leftrightarrow 526485 + 197485 = 723970$$

Ce qui identifie :

$$A = 4, B = 3, D = 5, E = 9, G = 1, L = 8, N = 6, O = 2, R = 7, T = 0$$

Il s'agit donc d'identifier pour chaque lettre et de manière unique, le chiffre qui vérifie l'addition. De manière générale, si l'on prend une addition avec n lettres différentes, l'espace de recherche a une taille de $\frac{10!}{(10-n)!}$. L'exemple précité qui contient 10 lettres distinctes a donc un espace de recherche de taille $10!$.

[Hogg et Huberman, 1993] proposent d'éliminer les recherches inutiles en permettant aux agents qui recherchent en parallèle une solution, d'échanger à l'aide d'un *tableau noir* des "astuces" ou solutions partielles qu'ils ont préalablement pu vérifier. Ces astuces consistent en des valeurs attribuées à des lettres qui vérifient une colonne au moins de l'addition avec ou sans retenue. Les agents restent libres d'utiliser ou non ces astuces grâce à une probabilité d'utilisation de celle-ci. Le tableau 5.1 rappelle les résultats de leurs expérimentations.

Problème	Rapport temps	$\frac{\text{TailleEspaceRecherche}}{\text{NombreSolutions}}$	% astuces membres d'une solution
$AB + AC = DE$	7	210	90-100
$WOW + HOT = TEA$	45	1844	50-60
$CLEAR + WATER = SCOTT$	145	181440	10-20
$DONALD + GERALD = ROBERT$	315	3628800	0,4

TAB. 5.1 – Résultats de [Hogg et Huberman, 1993].

Le rapport temps indique le rapport moyen entre le temps mis par un système non coopérant et un système coopérant à trouver une solution. On voit clairement que plus la complexité augmente, plus le rapport temps augmente malgré la difficulté grandissante à trouver une solution. Même si le pour-

centage des astuces utiles à une solution diminue, la coopération par distribution d'astuces entre les agents semble pourtant très efficace. En fait, les astuces permettent non seulement de trouver des combinaisons de valeurs utiles mais également d'éliminer rapidement les solutions inutiles. L'utilisation d'un tableau noir pour communiquer une partie du savoir acquis semble être une bonne méthode.

Leur deuxième expérimentation a consisté à étudier un problème de coloriage de graphes. La catégorie de problème étudié est considérée comme NP. Il s'agit de colorier un graphe composé de noeuds et d'arcs avec un minimum de couleurs de telle sorte qu'aucun noeud ne soit en contact par les arcs avec un noeud de même couleur. L'originalité de l'expérimentation vient de l'échange de savoir entre les agents. Cet échange consiste à écrire avec une probabilité q son état d'avancement dans le tableau noir et à lire avec une probabilité p une astuce compatible postée lorsque l'agent est bloqué et doit revenir en arrière dans sa recherche (*backtrack*). Une astuce compatible est donc un état de coloriage compatible avec celui en cours d'étude par l'agent qui lit l'astuce.

Lors de cette expérimentation, le gain en temps est moins significatif : il ne dépasse pas 2 en moyenne par rapport à une méthode heuristique classique de recherche (*Brelaz*). Les astuces postées ne sont qu'à 5% maximum des sous-ensembles de solutions finales. Pourtant les astuces postées ont permis encore une fois d'éviter des branches inutiles de recherche et ont permis d'améliorer la performance de la méthode Brelaz de manière significative.

Cette deuxième expérimentation montre encore une fois que pour échanger des connaissances, il faut qu'elles soient, ou bien interprétées de manière externe, ou bien compatibles avec l'agent.

En conclusion du travail de [Hogg et Huberman, 1993], on peut dire que l'utilisation d'un tableau noir symbolise un support de communication au travers duquel les agents communiquent et semble être une idée intéressante pour échanger de la connaissance. Toutefois, le savoir échangé peut être transformé comme dans la solution cryptarithmétique où on interprète ce qui est utile ou pas, à savoir une colonne d'addition juste. Cette solution est peu envisageable avec des agents simples car l'interprétation du savoir est indépendante d'eux. La deuxième solution consiste à rechercher des solutions compatibles ; c'est à dire à regarder si l'on peut faire évoluer l'état d'avancement de l'agent dans sa recherche sans pour autant le changer de branche de recherche. Cette solution, bien que plus dépendante de l'agent, oblige encore une entité extérieure au système multi-agents à contrôler que l'astuce fournie ne va pas éliminer une branche de recherche prometteuse.

Nous allons voir maintenant comment la communication par tableau noir peut également compenser un système de récompense local à travers l'étude de [Bull et al., 1995b, Bull et al., 1995a]. Il s'agit d'un robot équipé de quatre pattes appelé Gait qui doit apprendre à marcher à l'aide de système de classeurs de type Pittsburgh et d'un tableau noir. Chaque système de classeurs a en charge l'apprentissage de chaque patte. L'algorithme génétique sert à l'évolution. Les quatre pattes sont évaluées en parallèle et ne sont récompensées que lorsqu'elles réussissent à avancer. Pour cela la patte conditionnée reçoit en entrée l'information sur la distance restant à parcourir et sait donc si cette distance a diminué ou non. Chaque patte dispose d'un tableau noir et chaque classeur qui compose le système de classeurs de chaque patte a une partie communication qui sert de marqueur. Ce marqueur indique

le numéro de patte pour lequel l'information est destinée. Le message est posté dans le tableau noir correspondant et le marqueur est alors remplacé par le numéro de celui qui a envoyé l'information. Cette dernière consiste à indiquer à la patte receveuse, l'action qu'envisage de faire la patte émettrice afin que la première puisse adapter son action lors du prochain pas de temps. Les pattes sont évaluées pendant 20 pas de temps et l'on considère qu'il y a succès lorsque deux pattes au moins restent sur le sol afin d'assurer la stabilité du robot et que les autres donnent un coup d'avant en arrière pour faire avancer le robot. Il faut savoir qu'une patte du robot peut soit monter ou descendre, soit avancer ou reculer ou bien encore ne rien faire dans l'un des deux axes.

Le bilan de l'expérimentation de [Bull et al., 1995b] est intéressant :

The amount of communication varies but we have found legs at the same end of the robot, both the front and the back, passing messages to each other. These strings are then used by the receiver in the satisfaction of the conditions of its effector on the next time step. That is "contralateral" connections have been seen to emerge in the quadruped, which matches the biologists observations of natural quadrupedal systems...¹

La communication d'un savoir évolué a permis à un système multi-agents d'anticiper et de prévoir des actions pour mieux répondre au problème posé. En effet le robot arrive en moyenne à avancer dans 70% du temps et on dépasse les 87% du temps pour le meilleur apprentissage.

Au travers d'un problème prédateur-proies, [Edwin De Jong, 1999b] nous permet d'étudier comment le signalement des proies aux autres agents peut les aider à éviter des prédateurs. Cette communication s'effectue entre agents homogènes. Lorsqu'un agent voit un prédateur, il émet un signal qui va servir à la prise de décision des autres agents. Ce signal sert d'entrée supplémentaire dans l'établissement de situations type par chaque agent. Les agents vont apprendre à reconnaître, pour un signal émis, la position d'un prédateur en fonction de la leur, et ainsi éviter d'être capturés. Il est intéressant de noter que si on laisse suffisamment de temps aux agents pour apprendre à reconnaître la position des prédateurs, la communication est utile. Par contre, en dessous d'un certain seuil de visibilité du prédateur, la communication n'apporte rien. Il semble donc que la communication n'ait pas le temps d'améliorer le comportement des proies si elles ne peuvent apprendre à reconnaître les prédateurs. La proie ne s'est alors pas améliorée dans son comportement local habituel et ne bénéficiera pas d'une technique éventuellement apprise par une des autres proies pour éviter les prédateurs. Pour que la communication existe et récompense, il faut qu'elle soit utile. Ce constat peut sembler simpliste mais pose les limites de l'utilité de la communication. En effet, pour qu'il y ait communication, il faut qu'il y ait un phénomène répété qui doit être utile au groupe d'agents qui veulent communiquer. Si le phénomène, lié ici à un besoin de survie, ne se produit pas assez souvent, il ne justifie pas qu'on dépense de l'énergie pour l'identifier et modifier sa conduite. Nous avons déjà parlé de cela dans le chapitre 2, il est toutefois intéressant de noter que c'est la communication telle que nous venons de la

¹La part de communication fluctue mais nous avons pu constater que les pattes situées à la même extrémité du robot, à la fois l'avant et l'arrière, se passaient des messages entre elles. Ces chaînes sont alors utilisées par le receveur afin de satisfaire les conditions de ces effecteurs lors du prochain pas de temps. Ainsi, des connections "contralatérales" sont apparues dans le quadrupède, ce qui correspond aux observations faites par les biologistes sur des quadrupèdes naturels...

voir avec cet exemple qui rentre dans la catégorie des communications n'existant que ponctuellement. Pourtant cette communication apprise peut gagner à être incluse dans le comportement des agents si elle persiste.

Comme le souligne [Mataric, 1996] :

A group of agents learning in parallel creates a non-stationary world : as the agents learn, their behavior changes, resulting in inconsistencies.²

Nos expérimentations sur un groupe d'agents important l'ont confirmé : le comportement des agents qui se coordonnent n'est pas statique et par conséquent le monde qui est modifié ne peut l'être (cf. chapitre 4).

[Mataric, 1996] poursuit avec une autre difficulté :

Furthermore, multi-agent systems also face the credit assignment problem both at the level of the individual and at the level of the group. At the individual level, the interaction with other agents often delays an agent's payoff, aggravating the temporal credit assignment problem. At the group level, local individual behavior must be appropriately associated with global outcomes.³

Quelle que soit la solution que nous allons apporter pour distribuer la connaissance entre agents d'un système multi-agents, nous devons nous assurer tout d'abord que la connaissance transmise est utile au système aussi bien au niveau de l'agent que du groupe sans pour autant devoir interpréter et transformer la connaissance pour les agents. Le tableau noir semble être une solution pratique pour communiquer, il remplace avantageusement la description d'un médium de communication. Ensuite nous devons nous assurer que les agents s'améliorent aussi bien au niveau individuel que global.

5.3 Distribuer la connaissance

5.3.1 Cloner

[Quinn, 2000], à l'aide de robots Khepera, nous propose d'étudier une solution simple pour échanger de la connaissance. Il s'agit en effet ici de comparer deux approches pour communiquer un savoir entre deux robots afin qu'ils arrivent à se déplacer puis se suivre à la même vitesse pour éviter les collisions. La première approche consiste à littéralement cloner le meilleur comportement tandis que la seconde laisse évoluer séparément les deux robots. L'apprentissage se fait à l'aide de réseaux de neurones. Finalement c'est la deuxième solution qui triomphe de la première avec une majorité écrasante. En effet, cette solution obtient une performance de 95% tandis que la première approche obtient un

²Un groupe d'agents qui apprend en parallèle crée un environnement dynamique : tandis que les agents apprennent, leur comportement évolue, amenant à des inconsistances.

³De plus, les systèmes multi-agents se trouvent confrontés au problème de la récompense aussi bien au niveau de l'individu qu'à celui du groupe. Au niveau de l'individu, les interactions entre agents retardent la plupart du temps la récompense ce qui aggrave le problème de la répartition de la récompense dans le temps. Au niveau du groupe, le comportement local de l'individu doit être associé avec les buts du groupe de manière appropriée.

peu moins de la moitié de ces évaluations comme meilleur score. Pourtant cloner le meilleur semblait être une approche intéressante. Toutefois lorsque le problème de coordination spontanée est lié à une faible imprécision et une faible inertie apportée par le réseau de neurones, ce résultat semble cohérent. Le clonage du meilleur aurait plutôt tendance à augmenter l'inertie du système et progressivement la précision diminue comme nous l'avons vu au chapitre 4.

5.3.2 Application à un contrôle de carrefours routiers

Nous avons voulu définir un cadre homogène afin d'étudier comment échanger la connaissance. Ce cadre, nous l'avons trouvé en étudiant [Mikami et Kakazu, 1993, Mikami et Kakazu, 1994] qui nous ont permis de voir que le contrôle de carrefours routiers simples permettait d'étudier un système multi-agents homogène communicant tel que nous l'avons défini. Le problème du contrôle de carrefours routiers a été également et diversement abordé, nous ne citerons que l'approche *Genetic Programming* de [Montana et Czerwinski, 1996] et [Esczut et Fogarty, 1997] pour leur approche co-évolutionnaire. Leurs résultats sont encourageants mais ne permettent pas de répondre à la question de l'échange de connaissance dans un système multi-agents.

Le système multi-agents est composé de plusieurs carrefours routiers identiques connectés les uns aux autres et formant une grille. Le problème consiste à empêcher les embouteillages sur les carrefours tout en permettant aux automobilistes de traverser le plus rapidement possible le système vers leur destination. [Mikami et Kakazu, 1994] utilise une grille 3×3 pour leurs expérimentations. Chaque carrefour de la grille est contrôlé par un agent. Les agents sont homogènes i.e. ils sont physiquement identiques et ont le même rôle puisqu'ils doivent tous permettre aux voitures de traverser leur croisement le plus rapidement possible. Les carrefours ont une forme de croix et sont contrôlés par des feux bicolores synchronisés verticalement et horizontalement. Les voitures arrivent par les différentes routes des points cardinaux nord, est, sud et ouest. Lorsque les voitures s'approchent du croisement elle passe sur un premier capteur situé à distance du feu puis sur un second situé devant le feu (cf. figure 5.1).

Nous avons simplifié le modèle réel en donnant aux voitures une vitesse constante. Les voitures sont soit arrêtées au feu rouge, soit en train de rouler à vitesse constante : il n'y a ni accélération, ni décélération. Chaque route située sur le pourtour de la grille a une probabilité de flux de voitures qui indique la probabilité qu'une voiture apparaisse sur la route lors d'une évaluation. La file des voitures est limitée à 10 véhicules. A chaque début de cycle, le contrôleur décide si la couleur des feux doit changer ou non. Un cycle est composé de plusieurs pas de temps appelés *tics*. A chaque pas de temps, les véhicules qui peuvent avancer se mettent en mouvement, les autres attendent. Pour nos expérimentations, un cycle est composé de 5 tics. Un véhicule qui arrive à un croisement peut aller tout droit ou encore choisir de tourner à gauche ou à droite avec une probabilité de 10%. Une voiture qui traverse un croisement en tournant ou non arrive soit au croisement connecté correspondant, soit sort de la grille de carrefours. Nous supposons également qu'il n'y a jamais de collisions entre véhicules. Les routes sont à double-sens et les routes qui sont connectées à une autre intersection du système,

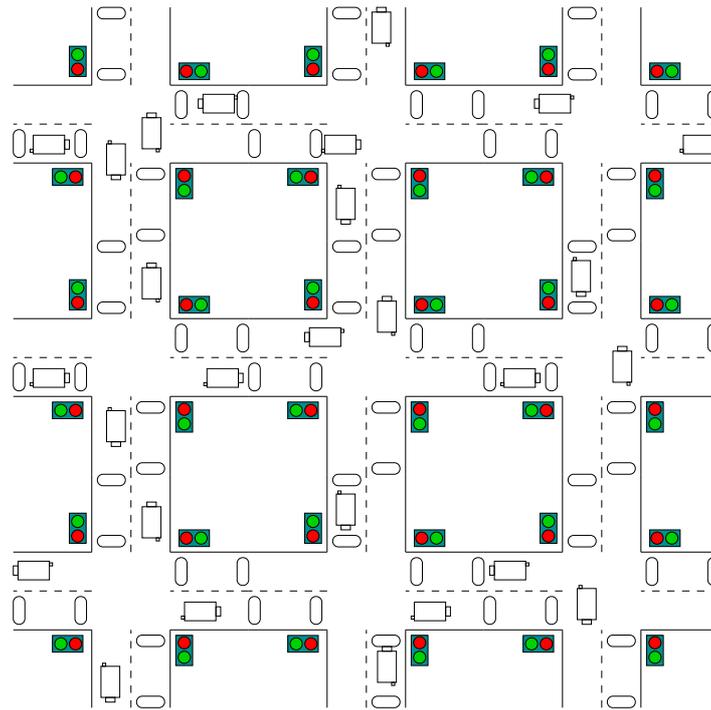


FIG. 5.1 – Grille 3 × 3 de carrefours.

reçoivent les voitures depuis cette intersection voisine i.e. il n’y a pas de probabilité de flux sur ces routes.

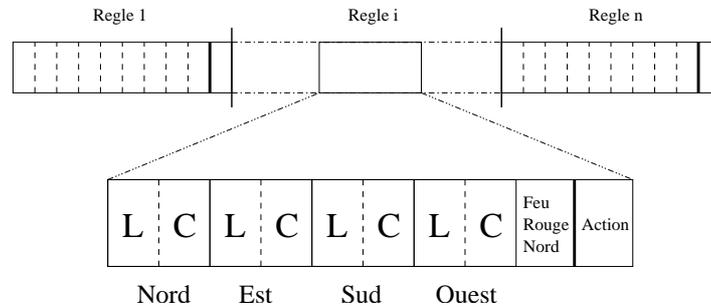


FIG. 5.2 – Un individu.

Les agents sont donc des contrôleurs composés chacun d’un système de classeurs de type Pittsburgh simplifié. Ils apprennent localement à résoudre le problème espérant ainsi améliorer la performance de l’ensemble. Les règles qui composent les individus de l’agent sont représentées dans la figure 5.2. Il y a donc quatre routes par croisement qui servent de signal d’entrée. Chaque route est elle-même composée de deux capteurs indiquant la présence (*vrai* ou *faux*) d’une voiture sur celui-ci : le capteur lointain (L) et le capteur proche du feu bicolore (C). Le dernier capteur utilisé est celui indiquant si le feu bicolore qui est sur la route nord est au rouge (*vrai*) ou au vert (*faux*). La couleur

des autres feux n'est pas utile puisque la couleur des trois autres feux peut se déduire directement de la couleur du feu bicolore qui est au nord. D'après notre description des capteurs, une règle de production est composée de $(4 \times 2) + 1$ soit 9 bits pour la partie condition et de 1 bit pour la partie action qui consiste à savoir si le contrôleur change (*vrai* ou *faux*) la couleur des feux bicolores de l'intersection. Ainsi, la règle 110011001 : 1 exprime le fait que lorsqu'il n'y a pas de voiture sur l'axe est-ouest du carrefour ($xx00xx00x : x$) et que l'axe nord-sud est saturé ($11xx11xxx : x$) et de plus que le feu de l'axe nord-sud est rouge ($xxxxxxx1 : x$), il est souhaitable de changer sa couleur ($xxxxxxx1 : 1$).

La récompense des agents est directement dépendante du temps d'attente moyen des voitures au croisement :

$$Recompense = \exp\left(\frac{\ln(0.7) \times \text{Delai_Actuel}}{\text{Premier_Delai}}\right)$$

Cette récompense est distribuée à la fin de chaque période d'évaluation et permet d'obtenir les courbes d'évolution qui se trouvent dans la figure 5.3.

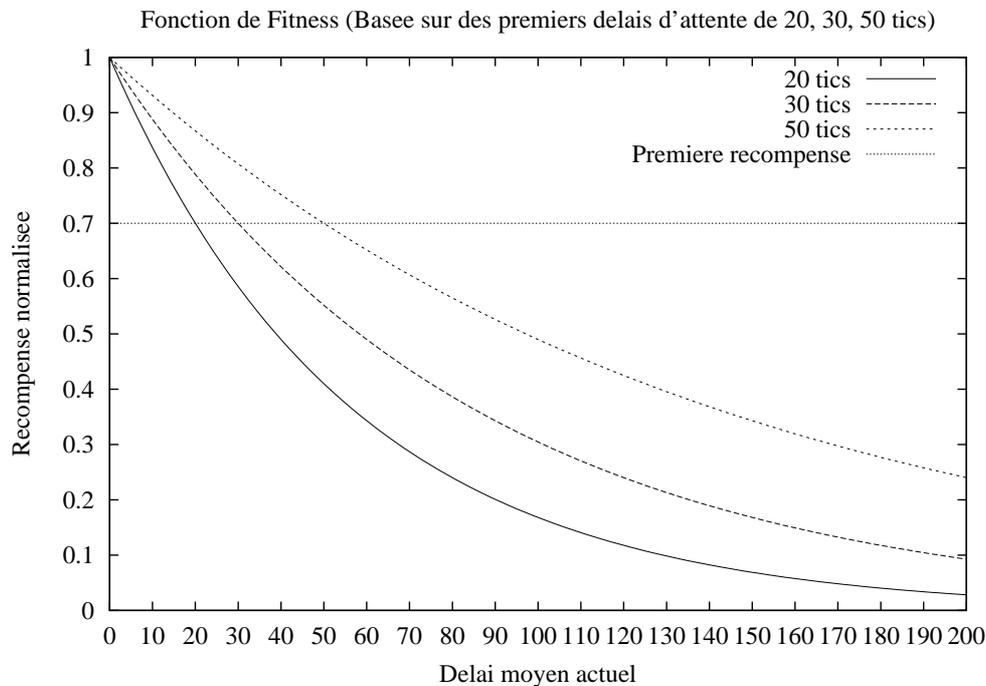


FIG. 5.3 – Fonction de Fitness.

Les agents de la première génération ont donc une fitness moyenne de 0.7 qui va évoluer en fonction des progrès qu'ils peuvent faire. La fonction exponentielle permet d'augmenter artificiellement l'attraction du système vers une diminution du délai.

Les paramètres de l'algorithme génétique se trouvent dans la table 5.3. Ils décrivent des paramètres toujours proches des valeurs rencontrées dans la littérature.

Le nombre de générations est faible à cause des réglages présentés dans la table 5.5 qui sont propres aux agents. 50 générations représentent en réalité :

Paramètre	Réglage
$P_{\#}$	40%
$P_{\text{Croisement}}$	60%
P_{Mutation}	1%
Mécanisme de Sélection	Roulette Wheel
Sélection des règles (cf. § 4.3.3)	Aléatoire
Croisement	Monopoint
Nombre de Générations	50

TAB. 5.3 – Paramètres de l’algorithme génétique.

$$50 \times 100 \times 5 = 25000 \text{ tics.}$$

Paramètre	Réglage
Nombre d’agents	9
Nombre d’individus (<i>NombreIndividus</i>)	20
Nombre de règles par individu (<i>NombreRegles</i>)	15
Nombre d’essais entre générations	100 cycles
Durée d’un cycle	5 tics

TAB. 5.5 – Paramètres du système multi-agents.

Nous avons réglé également les différents flux de voitures en fonction de probabilités qui sont représentées dans la table 5.6. Ces probabilités expriment le nombre moyen de voitures qui peuvent arriver sur une route tous les 60 tics. Une expérimentation avec des flux faibles sur les routes, devra contrôler 1250 ($25000 \times \frac{3}{60}$) voitures en moyenne tandis qu’une expérimentation avec des flux d’embouteillage devra permettre en moyenne à 12500 véhicules de passer.

Flux	Probabilité
Faible	$\frac{3}{60}$
Moyen	$\frac{6}{60}$
Élevé	$\frac{9}{60}$
Embouteillage	$\frac{30}{60}$

TAB. 5.6 – Flux des voitures sur les routes externes.

Nous avons moyenné toutes les expériences suivantes sur 20 expérimentations avec différentes racines aléatoires.

5.3.3 Élitisme

Élitisme standard

Notre première approche a été d'améliorer localement les agents. Pour cela, nous nous sommes inspirés du travail effectué avec les algorithmes génétiques par [De Jong, 1975]. Ce travail s'inspire de la théorie de Darwin sur l'évolution. En effet, Darwin nous dit que, d'après ses observations, l'évolution a toujours choisi les êtres les mieux adaptés et adaptables à leur milieu en éliminant ceux qui ne le sont pas. Cette idée s'applique aisément aux systèmes à base d'algorithme génétique puisque les individus qui composent un tel système disposent d'une force qui indique leur adaptation à l'environnement dans lequel ils évoluent. L'élitisme consiste à garder un certain nombre des individus, les mieux adaptés, à chaque génération, pour qu'ils puissent se diffuser progressivement dans la population. [De Jong, 1975] a ainsi appliqué l'élitisme aux algorithmes génétiques, démontrant l'efficacité de celui-ci. Le danger *a priori* de l'élitisme est de diriger le système vers une niche, ou optima local, d'où il ne pourra plus sortir. Nous avons voulu appliquer l'élitisme aux agents pour améliorer leur performance locale. Comme nous l'avons vu précédemment, chaque agent est composé d'un certain nombre d'individus qui sont évalués parallèlement. L'élitisme va consister à chaque génération à garder un certain nombre de ces individus qui ne subiront pas la pression sélective et seront présents tels quel à la génération suivante. La taille de la population du système étant constante, ces m individus sélectionnés vont naturellement remplacer les m individus les moins adaptés. Les individus restant sont traités alors par l'algorithme génétique et sont croisés et mutés.

Dans l'expérience que nous présentons maintenant, nous avons fixés les flux des routes de la grille 3×3 de carrefours comme suit :

- Routes au Nord : flux élevé.
- Routes au Sud : flux moyen.
- Routes à l'Est et à l'Ouest : flux faible.
- Routes connectées à un autre carrefour : flux nul puisque les voitures viennent de ce carrefour voisin.

Nous avons choisi différents pourcentages d'élitisme qui correspondent à différentes valeurs de m :

- 15% d'élitisme soit $m = 20 \times \frac{15}{100} = 3$
- 30% d'élitisme soit $m = 6$
- 50% soit $m = 10$
- 60% soit $m = 12$.

Toutes les courbes de résultats sont présentées avec une courbe de référence pour laquelle aucun élitisme n'a été appliqué. L'axe des ordonnées représente le temps moyen d'attente des voitures au croisement à chaque génération sur l'ensemble de la grille de carrefours et donne indirectement une évaluation de la fitness moyenne du système. Cette mesure est effectuée sur chaque véhicule afin d'estimer son temps d'attente au feu avant de pouvoir traverser.

La figure 5.4 montre les résultats expérimentaux. Tout d'abord, nous pouvons constater que sans élitisme, le système a besoin de plus de générations que n'importe laquelle des solutions avec élitisme

pour apprendre. L'élitisme semble permettre une convergence rapide et optimale du système. À la génération 10, les courbes ayant des pourcentages d'élitisme de 30%, 50% et 60% se rejoignent autour d'un délai d'attente moyen d'une voiture au croisement de 4.2 tics alors qu'il approchait les 48 tics au début de la simulation. A partir de la génération 10, l'effort pour baisser encore ce temps d'attente est important. Il faudra 40 générations supplémentaires pour atteindre un délai moyen d'attente de 1,9 tics pour les 3 courbes précitées. Les différences entre les courbes à 30%, 50% et 60% d'élitisme sont minimes. Le cas à 15% d'élitisme a plus de difficulté à être performant bien qu'il converge rapidement. Après analyse, le taux d'élitisme le plus performant et le plus rapide est de 50% d'élitisme. Ce taux permet une diminution de près de 96% du délai d'attente initial des véhicules. Un taux de 60% d'élitisme est trop élevé, la courbe qui représente les 60% se retrouve au dessus de celle des 50%.

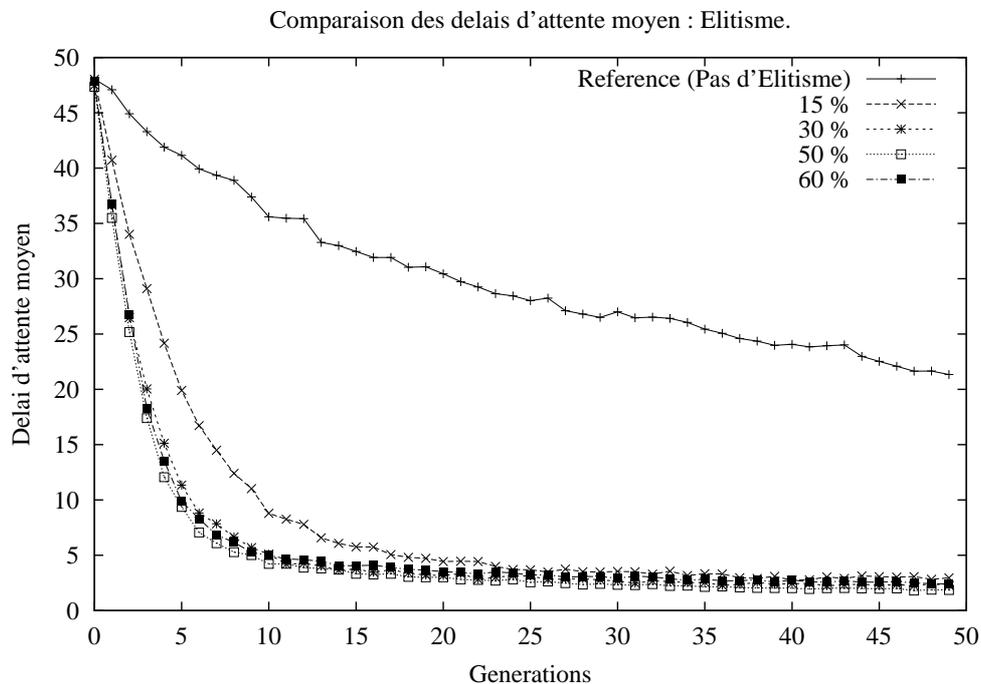


FIG. 5.4 – Élitisme.

Bien qu'élevé, le taux de 50% d'élitisme a permis d'améliorer considérablement la performance locale de chaque agent comme nous nous y attendions. Nous avons voulu voir s'il était possible d'améliorer cette méthode afin qu'elle permette d'encore améliorer la performance globale du système.

Élitisme distribué

L'environnement dans lequel nous travaillons est homogène, les agents travaillent en coordination. Ils sont tous physiquement identiques et travaillent sur le même type de problème de manière indépendante. Nous avons voulu permettre aux agents de partager leurs avancées dans leur apprentissage [Rosin et Belew, 1997]. Nous avons ainsi proposé l'*élitisme distribué* [Énée et Esczut, 1999]

qui consiste à échanger les meilleurs individus de chaque agent avec ceux des autres agents, dans le but d'améliorer l'ensemble du système. Cet échange se fait à travers une "réserve" qui va recevoir les individus en transit. L'élitisme distribué est basé sur une distribution équitable des agents. Cette distribution s'effectue après l'évaluation des individus et avant l'application de l'algorithme génétique (cf. algorithme 3).

Algorithme 3 Élitisme Distribué

Début

Remplir(P) { *Initialise aléatoirement la population initiale P .* }

$Generation = 1$

Répéter

Evaluer(P) { *Evaluer(P) consiste à évaluer l'ensemble de la population.* }

Appliquer_Elitisme_Distribué(P) { *Une fois l'évaluation terminée, on se sert des nouvelles "fitness" pour l'élitisme distribué.* }

AppliquerAG(P) { *Comme auparavant, c'est ici que l'élitisme simple est effectué si il y a lieu.* }

$Generation = Generation + 1$

Jusqu'à (Critères d'arrêt satisfait)

Fin

Le réservoir d'individus n'est pas un système de classeurs, il n'évolue pas et sert uniquement à échanger les individus. Le réservoir est rempli avec les meilleurs individus de chaque agent. Pour que l'échange soit le plus équitable possible, la force totale de chaque agent va permettre de connaître le nombre d'individus qu'il aura le droit de placer puis de prendre dans la réserve. Plus un agent a de force, plus il placera d'individus dans la réserve :

$$Nb_Mis_Reserve = Taille_Reserve \times \frac{FT(Agent)}{FT(Tous_Agents)}$$

FT est une fonction qui calcule la somme des forces d'un ou plusieurs agents. $Taille_Reserve$ est la taille de la réserve. Placer des individus dans la réserve ne signifie pas que les individus mis quittent l'agent pour être remplacés, un individu mis est un clone de l'original. Il s'agit ici de conserver le capital acquis tout en permettant aux autres agents de profiter des avancées que l'on a pu faire.

Une fois que la réserve est remplie avec les meilleurs individus de chaque agent, les individus de la réserve vont être redistribués aux agents en écrasant les plus mauvais individus qui les composent. La redistribution est ordonné : l'agent le plus mauvais i.e. celui avec $FT(Agent)$ le plus faible, prend les meilleurs individus de la réserve et le meilleur agent prend, quant à lui, les plus "mauvais" des meilleurs individus. Cette redistribution est donc de nouveau dépendante de la force globale de chaque agent et le nombre d'individus pris dans la réserve se calcule comme suit :

$$Nb_Pris_Reserve = Taille_Reserve \times \frac{FT(Tous_Agents) - FT(Agent)}{FT(Tous_Agents) \times (Nb_Agents - 1)}$$

Nb_Agents est le nombre d'agents du système soit 9. Le nombre pris dans la réserve est inversement proportionnel à celui qui est mis. Nous avons conservé les flux de l'expérience précédente. Nous comparons également les résultats obtenus avec ceux de la courbe de référence qui reflète une expérience sans aucun élitisme. Nous avons choisi différentes $Taille_Reserve$ qui se calcule en fonction du nombre d'individus de la population totale des agents (ici $3 \times 3 \text{ carrefours} \times 20 \text{ individus} =$

180 *individus*) et reflète un pourcentage de celle-ci : 15% soit 27 individus ($3 \times 3 \times 20 \times \frac{15}{100}$), 30% soit 54 individus et enfin 50% soit 90 individus dans la réserve.

La figure 5.5 décrit l'évolution du délai moyen d'attente au croisement avec l'élitisme distribué. Les résultats sont satisfaisants pour l'ensemble des expérimentations utilisant l'élitisme distribué. La performance générale est même plus élevée que pour l'élitisme simple. Voyons en détails les résultats. Les 3 courbes de l'élitisme distribué se rejoignent à la génération 15 avec un délai moyen d'attente de 3 tics pour les véhicules arrivant au croisement. Ce délai va progressivement diminuer pour atteindre 1 tic à la génération 50. Le meilleur résultat est l'expérimentation avec 30% d'élitisme distribué. En terme de performance pure, à la génération 50, la diminution du délai moyen d'attente est de l'ordre de 98% par rapport à la première génération. Mais il faut par contre 3 générations supplémentaires à l'élitisme distribué pour passer en dessous des 4.2 tics, soit 13 générations.

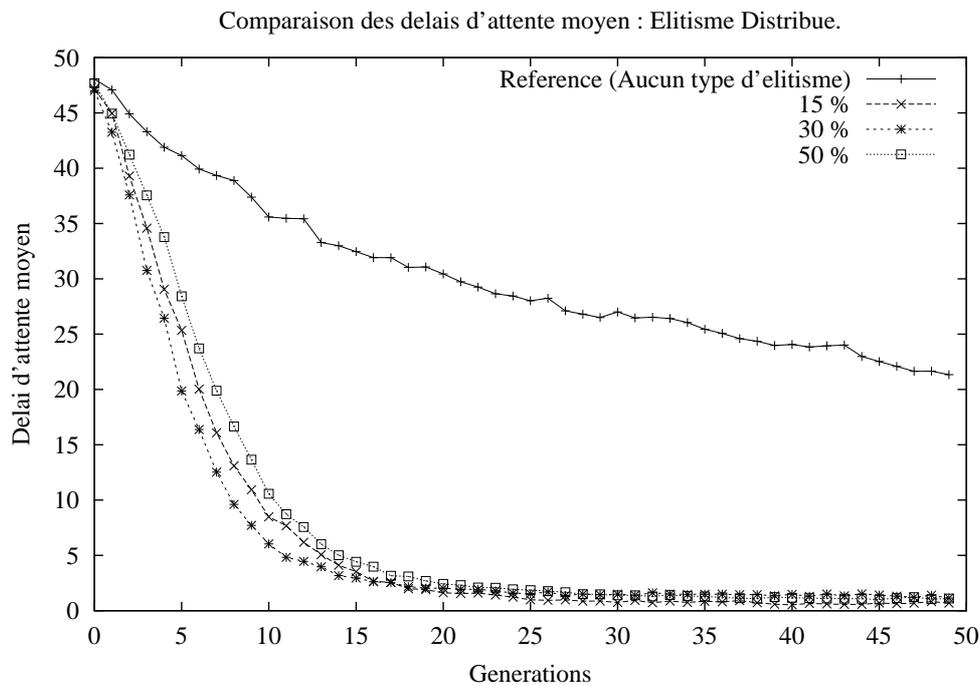


FIG. 5.5 – Élitisme Distribué.

Ce mécanisme d'échange des savoirs locaux vient de faire ses preuves et semble être une solution parfaitement adaptée aux systèmes multi-agents homogènes tels que nous les avons définis. Les meilleurs savoirs locaux sont échangés et se diffusent rapidement dans la population, améliorant la performance globale du système. Nous avons voulu tout de même voir si la combinaison d'élitisme standard et élitisme distribué pouvaient améliorer encore les performances du système.

Expérimentation combinée

Les deux méthodes d'élitisme ont montré leur efficacité, l'élitisme simple améliore la performance locale et le temps de convergence tandis que l'élitisme distribué améliore la performance globale du système. Nous avons donc réalisé une nouvelle expérimentation dans laquelle les deux méthodes sont combinées avec les meilleurs taux observés jusque là : 30% d'élitisme distribué et 50% d'élitisme simple. L'élitisme distribué est appliqué sur l'ensemble des agents et l'élitisme simple à chacun des agents séparément pendant l'habituelle phase de l'algorithme génétique. Nous espérons ainsi que le système va s'améliorer aussi bien au niveau local que global. Nous avons comparé les résultats avec 3 autres courbes :

1. La courbe de référence sans élitisme.
2. La courbe de l'élitisme distribué à 30% pour comparer avec le système le plus performant.
3. La courbe d'élitisme simple à 50% afin de comparer avec le système le plus rapide à converger.

Les résultats sont présentés dans la figure 5.6. Nous ne pensions pas que la diminution du délai moyen d'attente serait plus significative que lors des expérimentations précédentes. Pourtant, force est de constater qu'à la cinquième génération le délai d'attente moyen des véhicules au croisement n'est déjà plus que de 3,4 tics devant l'élitisme simple pour la vitesse de convergence. Non seulement le système converge beaucoup plus vite mais il est également plus performant puisqu'à la génération 50, on obtient un délai moyen d'attente de 0,47 tics soit une diminution de 57,27% de ce délai par rapport à l'élitisme distribué à 30% à la même génération.

Pour résumer, les voitures ont la plupart du temps le feu au vert pour traverser les différents carrefours malgré des flux dominants importants. Afin de valider notre approche, nous avons voulu tester la robustesse du système en faisant varier le flux de voitures à intervalles réguliers. Nous avons donc fait 3 expériences supplémentaires que nous présentons maintenant.

5.3.4 Robustesse

Les deux premières expériences que nous avons réalisées, consistent à changer le flux des voitures toutes les 10 générations jusqu'à la génération 40. À partir de la génération 40, le flux est choisi aléatoirement. Dans la première expérience, nous avons étudié la réaction du système face à une augmentation progressive du flux de voitures. Dans la seconde, nous avons diminué progressivement le flux de voitures. Les flux vont varier de faible à embouteillage et inversement tels que nous les avons décrits dans la table 5.6. Un flux moyen de 30 voitures par 60 tics est une situation qui devrait créer d'importantes difficultés au système.

Pour la première expérience, le flux est d'abord faible pour toutes les routes lors des 10 premières générations puis devient moyen pour les 10 suivantes. Il est élevé pour les générations 20 à 29 et enfin passe à l'embouteillage pour les 10 générations avant 40. La figure 5.7 présente les résultats avec une courbe de référence sans élitisme et une courbe avec la solution combinée ayant les mêmes taux qu'auparavant. Lors de cette expérimentation, les délais de départ sont différents avec environ 60 tics pour

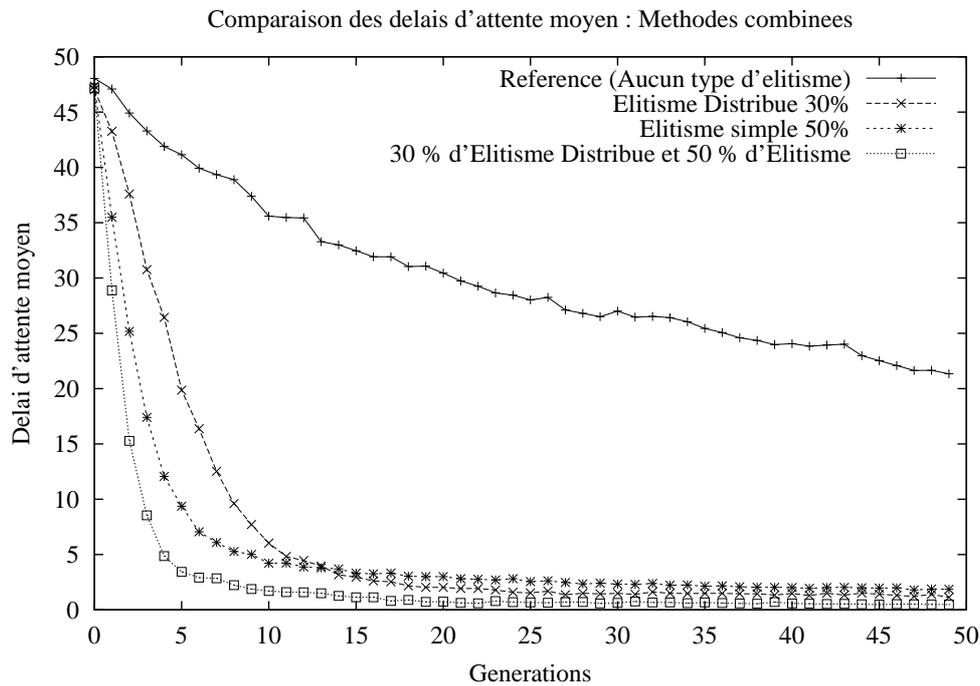


FIG. 5.6 – Expérimentation combinée.

la courbe de référence et 40 tics pour la solution combinée. Bien qu'ayant considérablement amélioré sa performance, l'expérimentation de référence reste toujours en retrait avec 20 tics environ à la fin de la génération 50. Si l'on regarde de plus près la solution combinée, on s'aperçoit qu'à chaque augmentation de flux, le système doit apprendre à maîtriser la nouvelle difficulté. Pourtant, le système converge parfaitement malgré une grosse difficulté lors du passage à un flux d'embouteillage à la génération 30. Les individus qui composent les agents semblent parfaitement capables de s'adapter aux nouvelles difficultés, comme si ils disposaient en permanence des individus et classeurs nécessaires pour répondre aux sollicitations des nouveaux flux de voitures. Enfin la courbe de référence est lisse car elle n'a pas eu le temps d'apprendre à résoudre la difficulté précédente qu'une nouvelle commence, il s'agit donc d'un apprentissage continu dans ce cas. Comme nous nous y attendions, à partir de la génération 40, gérer n'importe lequel des flux précédemment rencontrés ne pose plus de problème à la solution combinée. Elle vient de montrer sa robustesse et sa capacité d'adaptation, la population d'agents conserve donc toujours la possibilité d'apprendre de nouvelles situations.

Nous allons voir à l'aide de la seconde expérimentation si le fait d'apprendre d'abord les choses difficiles avantage la solution combinée. Le flux de voitures varie d'embouteillage à faible par pas de 10 générations. Les 10 dernières générations se font avec un flux aléatoire comme pour la première expérimentation. La figure 5.8 présente les résultats obtenus. Cette fois encore la courbe de référence évolue à l'opposé de celle de la solution combinée. En effet, toutes les 10 générations, on assiste à un sursaut de difficulté pour la courbe de référence. Ce résultat contre-intuitif semble lié au fait

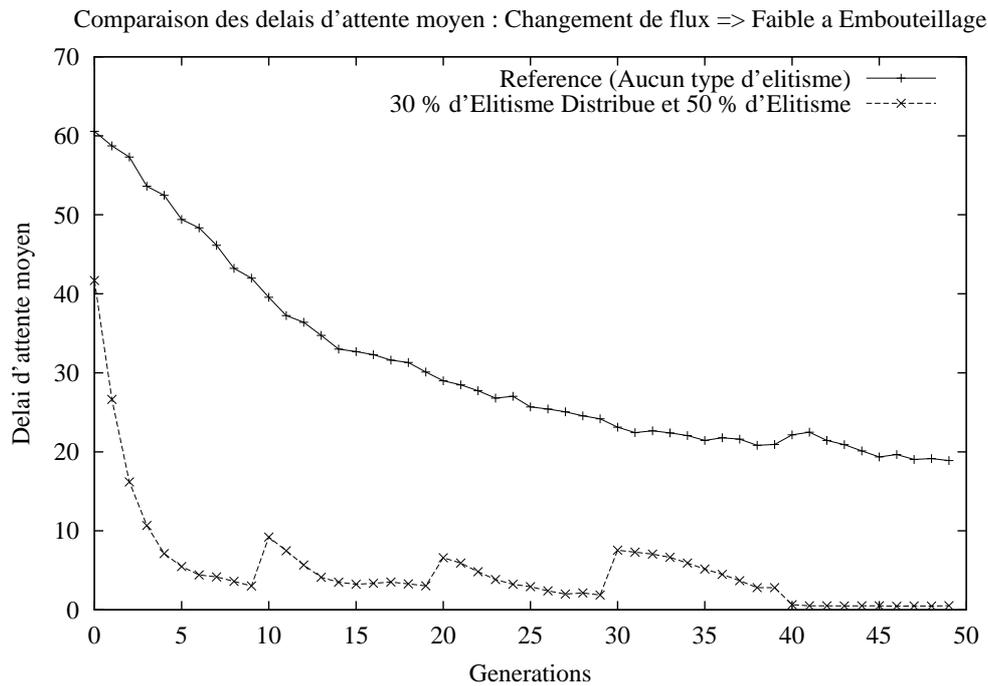


FIG. 5.7 – Flux croissants.

que le système n'ait pas encore convergé au moment du changement de flux, il est donc incapable d'anticiper la diminution de flux car ces individus ne sont pas parfaitement adaptés à la situation précédente, plus difficile. Revenons maintenant à la courbe de la solution combinée. Nous constatons que d'apprendre en premier à gérer un flux de voiture très important avantage considérablement la solution combinée. En effet, elle apprend légèrement plus vite à gérer la difficulté que la facilité. Nous obtenons ensuite un comportement qui semble logique : une fois que l'on sait parfaitement gérer une situation difficile, les autres situations sont facilement maîtrisées. Pendant les 40 générations restantes, la solution combinée maintient un délai d'attente très bas au carrefour. On peut supposer que les individus contenant les règles permettant de résoudre un problème de flux importants contiennent également les règles permettant de résoudre un problème plus facile. La robustesse et l'adaptation de la solution combinée a été montrée dans les deux sens. Nous avons voulu vérifier à l'aide d'une dernière expérimentation que l'apprentissage offert par la solution combinée était définitivement robuste.

Cette dernière expérience consiste à changer aléatoirement le flux des voitures toutes les 5 générations pendant les 50 générations de l'expérimentation. Nous présentons les résultats à l'aide de la figure 5.9. Nous constatons que la courbe de référence progresse comme précédemment vers les 20 ticks à la génération 50 tout en faisant de légers bonds toutes les 5 générations. La solution combinée quant à elle, diminue rapidement le délai d'attente moyen bien qu'il y ait un sursaut à la génération 5 qui semble indiquer que le système n'a pas encore complètement résolu le problème. Cette hypothèse se confirme puisque la courbe reste pratiquement lisse par la suite jusqu'à la dernière génération. Nous

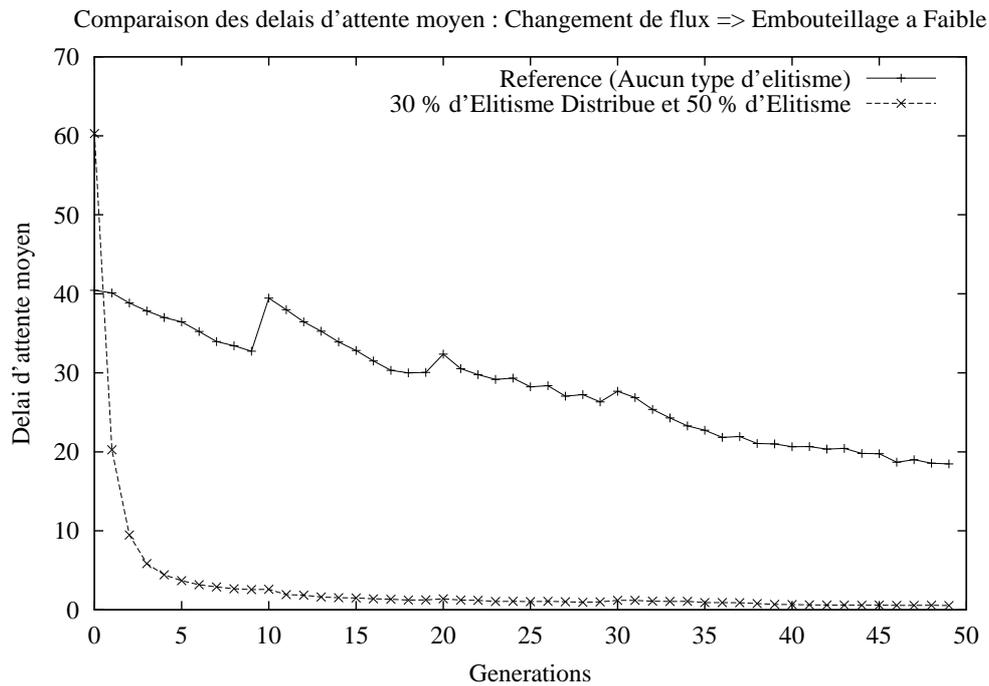


FIG. 5.8 – Flux décroissants.

avons pu observer qu'après la génération 15, les sursauts toutes les 5 générations ne dépassent pas les 0,2 tics. De nouveau, nous mettons en évidence la robustesse de la solution combinée avec au final un délai d'attente moyen des véhicules de 0,57 tics aux croisements.

5.4 Conclusion

Nous avons montré dans ce chapitre que pour les systèmes multi-agents homogènes communicant leur savoir, les systèmes de classeurs de type Pittsburgh simplifié offraient une plate-forme adaptée à l'échange de connaissances. A partir de constats simples, nous avons pu nous rendre compte que de tels systèmes devaient pouvoir échanger totalement la connaissance entre les agents sans pour autant que la solution ne soit trop coûteuse. Nous nous sommes inspirés de solutions améliorant localement les agents afin de trouver une solution permettant d'améliorer le système dans sa globalité. Nous y sommes parvenus en appliquant l'élitisme à la population entière des agents, créant ainsi l'élitisme distribué qui consiste à échanger à l'aide d'une réserve, les meilleurs individus des agents de manière équitable et ce en remplaçant les individus les moins performants. Cette solution n'a pu être appliquée que parce que les systèmes de classeurs de type Pittsburgh simplifié ont cette particularité d'être composés d'individus qui eux-mêmes contiennent des classeurs qui participent à la solution potentielle du problème. Nous avons vu que l'élitisme distribué améliorait notablement la performance du système en dépit du temps de convergence plus grand que pour l'élitisme simple. C'est alors que nous

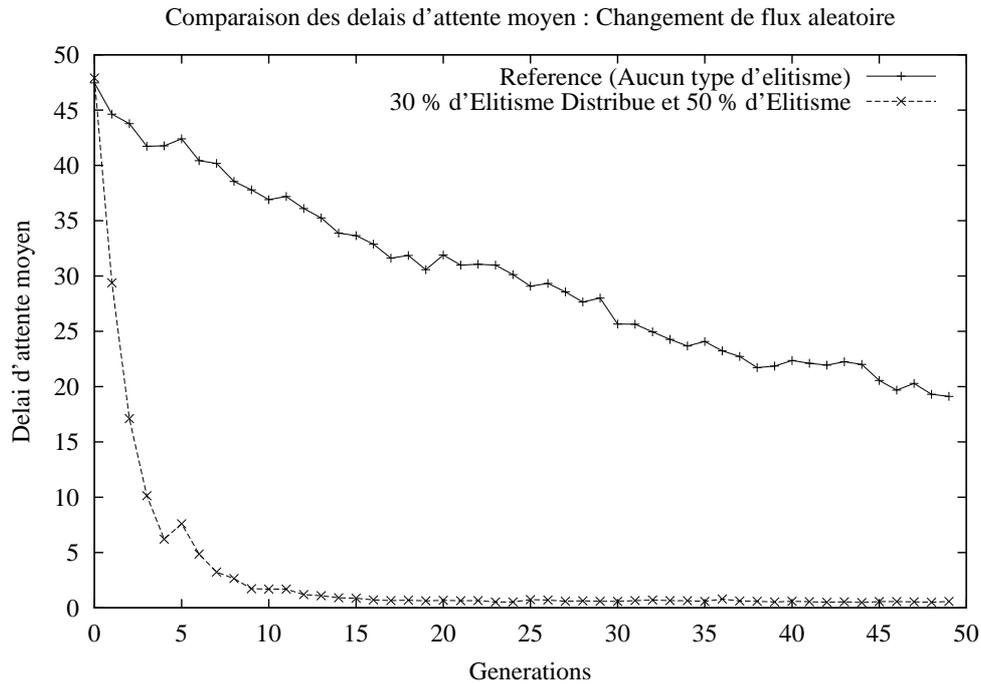


FIG. 5.9 – Flux aléatoires.

avons décidé d'utiliser ces deux méthodes afin de faire progresser aussi bien localement que globalement le système multi-agents attelé à la résolution d'un problème de gestion d'une grille de 3×3 carrefours routiers en forme de croix. La solution combinée a prouvé sa supériorité et nous avons également montré sa robustesse. Nous avons donc obtenu une solution simple et efficace que l'on peut facilement appliquer aux systèmes multi-agents homogènes cherchant à communiquer un savoir. Nous avons donc étudié jusqu'à présent les limites de la coordination des systèmes multi-agents et nous avons vu comment échanger un savoir dans un cadre homogène. Nous allons maintenant voir quels sont les problèmes que peuvent engendrer l'hétérogénéité lorsque l'on cherche à échanger des informations à l'aide d'un système multi-agents.

Chapitre 6

Communiquer avec des SMA Hétérogènes

Nous avons proposé des solutions afin que les agents, tels que nous les avons définis, puissent se coordonner ou bien encore échanger un savoir en le communiquant aux autres membres du groupe homogène auxquels ils appartiennent. Il nous reste à étudier et comprendre comment nous allons pouvoir permettre à ces agents hétérogènes d'échanger des informations. Dans un premier temps nous allons expliciter la notion d'hétérogénéité afin de donner, dans un second temps, un cadre à l'étude d'un modèle minimal de communication. Une fois le modèle décrit, nous testerons sa validité sur différents systèmes de classeurs puis nous l'appliquerons à un jeu de position pour des robots Khépéra simulés. Enfin, nous vérifierons sa validité en étendant le modèle. Cette dernière étude fera l'objet de diverses expérimentations permettant de mieux comprendre la dynamique des systèmes multi-agents hétérogènes communicants.

6.1 Problématique des SMA hétérogènes

Revenons un instant à la fourmi fourragère. Lorsqu'elle émet un signal d'alarme, que fait-elle réellement ? Quel est le contexte exact ? Quelle est la réponse ? Qui est concerné par la réponse ? Comment les autres fourmis réagissent-elles à cette réponse ?

Nous allons prendre le temps de répondre à ces quelques questions car la fourmi est un modèle simple, adaptable aux agents tels que nous les avons définis précédemment. Voici donc une fourmi ouvrière qui rencontre une autre fourmi qui n'appartient pas à la colonie, ses capteurs olfactifs lui indiquent immédiatement que cette fourmi n'a pas été élevée dans la colonie. Son instinct lui indique qu'il faut informer le plus rapidement toutes les fourmis de sa colonie du danger potentiel que représente cette fourmi étrangère. Pourquoi ? Simplement parce que des milliers d'années d'évolution ont fait intégrer dans le capital génétique de la fourmi que tout être qui est étranger à sa colonie est probablement dangereux. Cette réaction est typiquement le résultat d'une évolution Baldwinienne dans laquelle l'acquis durant le vivant, ici le danger, a été intégré au capital génétique pour devenir instinct.

Bien qu'extrêmement protectionniste, ce comportement de la fourmi ouvrière donne une chance à ses congénères de se défendre.

En résumé, la fourmi ouvrière va émettre un signal dans une "langue" que les fourmis comprennent, à savoir un signal chimique qui peut permettre en plus d'indiquer la distance à laquelle se trouve le danger. Ce signal va concerner toutes les fourmis de la colonie, il y a donc une appartenance à un système multi-agents qui justifie ce signal. Pourtant les réactions seront diverses au sein de la colonie. Les fourmis ouvrières membres de la colonie et proches du signal vont soigneusement éviter l'endroit et vont éventuellement amplifier le signal si elles rencontrent elles-même l'intruse. Si ces fourmis se trouvent trop près de la source du signal, elles se prépareront également à attaquer. Les fourmis soldats vont, quant à elles, aller directement vers la source du signal pour prêter main forte aux ouvrières en détresse. Dans les deux cas, le signal émis est une transposition simplifiée de ce qu'ont pu lire les capteurs de la fourmi émettrice. Cette transposition est simplifiée puisque les capteurs sensoriels de cette fourmi peuvent rencontrer une large gamme de valeurs qui correspondent à un signal émis. Ce signal simplifie donc l'information première reçue au risque de méprendre les autres fourmis. C'est pour cette raison qu'il y a des fourmis soldats qui sont spécialisées dans la lecture de cette catégorie de signaux et qui vont aller sur les lieux du signal pour avoir leur propre interprétation de l'information. Sans entrer dans les détails de la chaîne des réactions qui vont amener à l'éradication de l'intruse, on comprend qu'il existe différentes catégories de fourmis. Chaque catégorie forme un système multi-agents homogènes communicant ou non et l'ensemble des catégories forme une colonie de fourmis, à savoir un système multi-agents hétérogènes communicant selon ses besoins.

Dores et déjà nous constatons que pour que des agents hétérogènes communiquent, il faut qu'il y ait une nécessité [Di Paolo, 1998]. La communication hétérogène a donc un coût qui est moindre que l'homogénéisation totale des agents car leur spécificité est nécessaire à la colonie. Mais ce coût reste élevé puisque les agents apprennent à ne pas émettre de signaux inutiles. Nous reparlerons de ce point un peu plus loin.

Nous avons vu avec les fourmis l'utilité d'un langage pour communiquer entre les différentes communautés qui composent la colonie. Par contre, nous n'avons pas cerné comment un tel langage a pu apparaître et évoluer. Enfin nous n'avons pas défini ce qu'est un langage.

Un langage se compose en règle générale d'un lexique, à savoir le vocabulaire, et d'une grammaire qui permet d'articuler le vocabulaire pour former des phrases. Nous ne rentrerons pas ici dans l'aspect syntaxique. Un lexique est un tableau à 2 dimensions de correspondances entre chaque mot du vocabulaire et sa signification. La formation d'un lexique est la première phase de compréhension d'un langage. La grammaire constitue un deuxième temps dans l'apprentissage. Comme l'indique [Edwin De Jong, 1998], le fait qu'un agent utilise un mot d'un langage n'implique pas qu'il l'interprète de la même manière que d'autres agents. Il suggère de plus, que les agents adaptent leur communication durant leur vie (ontogenèse) et pas durant leur évolution (phylogénèse).

A ce propos, de nombreuses théories se sont opposées quant à la façon dont le langage humain est implanté. Comme nous l'avons vu, deux théories émergent. La première défend l'idée que le lan-

gage est une composante génétique intégrée (phylogénique) : le Language Acquisition Device (LAD), qui contient les bases d'une grammaire universelle, permet d'apprendre n'importe quelle langue. La seconde montre que l'apparition du langage est purement le fait d'un apprentissage (ontogénique) et qu'il n'y a pas de fonction spécialisée du cerveau pour le langage.

[Briscoe E. J., 1999b] a effectué une expérience basée sur le LAD. Son approche utilise un système basé à la fois sur l'évolution pour les fondements grammaticaux des langues, à savoir la grammaire universelle, et à la fois sur l'apprentissage qui consiste à adapter le LAD à un langage auquel il est exposé. Le LAD évolue parce qu'il est exposé à différents langages mais aussi parce que les langages évoluent. Cette approche de l'assimilation génétique est typiquement l'effet Baldwin. On est en droit de se demander alors pourquoi le LAD n'a pas intégré dans ses gènes un langage universel ? En fait la réponse se fait en trois étapes extraites de [Briscoe E. J., 1999b] :

1. Waddington (1975 :307) remarks : 'Evolution is quite capable of performing such a feat... But in the case of language, there is certainly little reason to see why it would have been advantageous to press the matter further. If a child which had never met a language-user developed the ability to talk, who after all would it talk to ?'¹

2. Deacon ... suggests, in particular, that languages have evolved to be easily learnable by an acquisition procedure which 'starts small', following Elman (1993) ..., with a limited working memory only capable of 'seeing' local grammatical dependencies.²

3. The speed at which linguistic changes can diffuse through a population will be far faster than that at which genetic change can do so. However, there is clearly a speed limit to this change within a successfully communicating population, and that speed limit means that only a small part of the space of possible grammars may be sampled over the period required for biological evolution. The experiments reported ... suggest this can lead to a constant enough selection pressure capable of supporting genetic assimilation of a LAD. However, the fact of linguistic change provides a natural barrier to total genetic assimilation of a fully-specified grammar.³

Il paraît maintenant établi que le langage se construit depuis la naissance et 'commence petit' par la construction d'un lexique. Ce lexique est impératif pour permettre d'identifier de manière précise chaque élément de l'environnement des agents communicants. Le fait que les langages soient

¹Waddington note : 'L'évolution est parfaitement capable d'en arriver là... Mais dans le cas du langage, il y a certainement peu de raisons de voir en quoi il aurait été avantageux d'aller plus loin. Si un enfant qui n'a jamais rencontré un être doté d'un langage avait développé la capacité de parler, à qui parlera-t-il finalement ?'

²Deacon ... suggère, en particulier, que les langages ont évolué pour être facilement appris en utilisant une procédure d'acquisition qui 'commence petite', suivant Elman (1993) par cette idée ..., avec une mémoire limitée seulement capable de 'voir' les dépendances grammaticales locales.

³La vitesse à laquelle se diffuse les changements linguistiques dans une population sera toujours plus rapide que n'importe quelle évolution génétique ne peut le faire. Pourtant, il y a une vitesse limite à de tels changements dans une population communicante, et cette vitesse limite implique que seule une petite partie de l'espace des grammaires possibles peut être échantillonnée pendant une génération d'évolution. Les expériences faites suggèrent que ce phénomène entraîne une pression sélective constante rendant acceptable l'idée d'assimilation génétique du LAD. Quoi qu'il en soit, les changements linguistiques fournissent une barrière naturelle à toute assimilation génétique complète d'une grammaire également complète.

en constante adaptation entraîne deux phénomènes clairement exposés par [Briscoe E. J., 1999b] : la vitesse d'adaptation empêche l'évolution de fournir un langage complet dès la naissance et par opposition, on ne peut pas parcourir entièrement l'espace de recherche que représente l'ensemble des grammaires possibles dans un temps trop court. Toutefois il omet de dire que puisque le temps est compté dans la vie d'un agent, la communication a un coût non négligeable et que par conséquent les agents (les humains ici) cherchent à diminuer ce coût en ne parcourant pas l'ensemble des grammaires possibles. [Kaplan, 2001]⁴ va même plus loin à l'aide d'une expérimentation en montrant que cette optimisation du temps ne se fait pas au détriment du sens et donc que les agents choisissent toujours une solution ayant une taille adaptée à leur capacité : ni trop courte (pas assez de sens exprimables), ni trop longue (trop de sens exprimés) pour éviter d'introduire du bruit dans le lexique formé.

Lors de ces expériences, [Briscoe, 1999a] sépare les grammaires des différentes langues humaines en groupes ayant différentes dérivations de l'ordre des groupes "sujet" "verbe" et "complément". A partir de deux grammaires contenues dans le LAD et d'un mécanisme simple d'acquisition, il arrive à apprendre l'ensemble des grammaires des différentes langues. Il constate aussi que sous certaines conditions, un agent apprenant peut changer la grammaire qu'il acquiert, justifiant ainsi que le mécanisme de base de l'apprentissage est bien le LAD qui fournit les briques de base à l'acquisition d'un langage évoluant.

[Steels, 1996b] défend la seconde idée :

There is no evidence that a *particular* natural language is innate (although the thesis has been advanced that there is an innate language acquisition device).⁵

Il montre même que des agents sont capables de créer un lexique commun en s'auto-organisant. Il explique ce phénomène comme suit :

Each agent is assumed to create his own vocabulary in a random fashion. But agents are coupled because they must share vocabularies in order to obtain the benefit of cooperating through communication. Agents therefore keep changing their own private vocabulary until it is conform to the common vocabulary.⁶

On peut faire apparaître un lexique commun cohérent à partir d'agents aux langages hétérogènes grâce à l'apprentissage. Nous venons de voir que, dans le cas particulier des êtres humains, la formation d'un langage est une chose complexe qui commence par la formation d'un lexique puis l'acquisition d'une grammaire. Nous ne souhaitons pas aborder l'aspect grammatical de la formation d'un langage car les agents que nous utilisons, sont plus simples que des fourmis et ne disposent que de capteurs de taille très limitée. Ce constat permet de mieux comprendre d'où peut provenir le besoin de communiquer lorsque les agents sont hétérogènes. En effet, des agents d'une communauté se sont spécialisés

⁴Pages 81 à 86.

⁵Il n'existe aucune preuve qu'un langage naturel particulier soit inné (Bien qu'il existe une thèse selon laquelle il existerait un système d'acquisition du langage).

⁶Chaque agent a la possibilité de créer de manière aléatoire son propre vocabulaire. Mais les agents sont rassemblés car ils doivent partager leurs vocabulaires afin de bénéficier de la coopération par communication. Les agents adaptent donc leur vocabulaire privé jusqu'à ce qu'il soit conforme au vocabulaire commun.

au cours de l'évolution dans certaines tâches formant ainsi des catégories d'agents. Une catégorie d'agents représente un domaine d'expertise de l'environnement [van Eijk et al., 1998]. Faire partie d'un domaine d'expertise oblige un groupe d'agents souhaitant rester en communauté à trouver un référent commun que les autres groupes d'agents de la communauté puisse comprendre. De plus, pour que la communication se produise entre ces différentes catégories d'agents, il est nécessaire que la communication modifie également les convictions des agents sur le comportement à avoir face à un environnement donné. La spécialisation d'un groupe d'agents dans une communauté les oblige donc à trouver un moyen de communiquer. Créer un super-individu a un coût trop important étant donné les limitations naturelles de chaque agent. La complexification des espèces ne passe donc plus seulement par l'intégration génétique de celle-ci mais par une étape suivante de l'évolution qui est de permettre au vivant d'apprendre et d'échanger, c'est l'effet Baldwin.

Nous allons étudier plus avant sur les problèmes que les agents "simples" rencontrent lors de la formation d'une "langue" commune. [Edwin De Jong, 1999a] nous définit la communication ainsi :

Once agents have concepts at their disposal that capture relevant situations in the environment, a basis for developing communication is present. At each timestep, every agent produces a signal corresponding to its perceived situation. When some agent does not receive sensory information that would allow it to determine its situation, this information may be obtained through communication. If the association between a signal received and a situation concept other than the one indicated by the sensors is strong enough, the agent can deduce its actual situation from that signal and act accordingly.⁷

Il faut donc qu'il y ait nécessité de communiquer et plus particulièrement, qu'il y ait nécessité de communiquer ce que l'on perçoit pour que la communication apparaisse au sein des systèmes multi-agents. Cet échange s'apparente aux *Language Games* proposés par [Steels, 1996c]. Le but de ces jeux est de permettre à des agents de définir un lexique commun pour identifier ce qu'ils perçoivent. Ce lexique est défini par des mots tirés au hasard et des sens clairement identifiés. Un sens peut être la désignation d'un objet ou d'un autre agent. [Steels, 1996b] montre que des agents sont capables de créer un lexique commun pour désigner ce qu'ils perçoivent par le seul fait de l'auto-organisation. Bien évidemment, le système passe par une phase de confusion durant laquelle les différents agents essayent d'imposer leur choix lexical. A la fin de l'expérimentation pourtant, on retrouve un système parfaitement organisé et capable de dialoguer. Les expérimentations de [Edwin De Jong, 1999a] sur le même thème ont montré que lorsqu'un mot n'est pas utilisé par suffisamment de monde, celui-ci finit par être oublié ce qui justifie parfaitement les résultats obtenus par [Steels, 1996b].

Pourtant comme nous l'avons souligné auparavant, et comme [Steels, 1996a] le rappelle :

⁷Une fois que les agents ont des concepts (sens) à leur disposition qui relèvent de situations réelles de l'environnement, une base de développement de la communication est présente. A chaque pas de temps, tous les agents produisent un signal adapté à la situation qu'ils perçoivent. Lorsque certains agents ne disposent pas d'informations sensorielles qui leur permettent d'appréhender leur situation, cette information peut alors être obtenue par communication. Si l'association entre un signal reçu et une situation autre que celle perçue par les capteurs est suffisamment forte, l'agent peut déduire sa situation actuelle à l'aide de ce signal et prendre les décisions qui s'imposent.

1. The same word may have different meanings, and therefore an expression may be ambiguous. The context or additional words are then used to determine which meaning is intended.⁸

2. The same meaning may be expressible by different words, and therefore we see synonyms emerge.⁹

3. There is never complete coherence among agents due to different developmental pathways. Nevertheless almost complete communicative success is observed after a sufficient number of language games, because the language has become adequate enough to deal with all possible contexts of use.¹⁰

Tout langage possède des mots homonymes qui expriment plusieurs sens, ou encore plusieurs mots synonymes qui expriment un seul sens. Le jeu du langage impose donc d'étudier à la fois le succès en communication mais également la formation du lexique et dans ce lexique de regarder la part d'homonymie et de synonymie.

L'étude de la formation d'un lexique est une première étape nécessaire pour vérifier l'aptitude des agents hétérogènes à communiquer. Les agents devront pour se faire être capable d'identifier clairement ce qu'un autre agent observe à l'aide de mots. Le "langage" qui va émerger dans un premier temps, doit être étudié en définissant des mesures de succès de la communication. Dans un deuxième temps, nous devons définir des mesures d'homonymie et de synonymie. Nous allons devoir également définir comment les systèmes de classeurs peuvent apprendre et intégrer l'effet Baldwin qui semble être une clé de l'évolution.

6.2 Définition d'un modèle minimal

Nous avons choisi d'étudier la communication sous sa forme la plus simple possible afin d'en dégager les principales caractéristiques. Les systèmes de classeurs sont des outils d'apprentissage composés de classeurs qui sont eux-même l'expression de ce que reçoit comme information le système sur son environnement et l'action qu'ils peuvent envisager de faire. Ces systèmes purement réactifs apprennent à s'adapter aux cours du temps à l'environnement dans lequel ils évoluent. Si l'on se réfère à l'effet Baldwin, les classeurs doivent être le fruit de l'évolution et de l'apprentissage. La communication a lieu pendant la vie des agents (ontogenèse). Elle est séparée de l'évolution normale des agents (phylogenèse). Par conséquent, nous étudierons la communication durant la vie des agents avant qu'eux-même n'évoluent. Nous avons également choisi de n'étudier que la communication en appliquant des techniques d'apprentissage pour permettre aux agents de faire "évoluer" la communication comme l'a proposé [Steels, 1997].

⁸Le même mot peut avoir plusieurs significations, et par conséquent, une expression peut être ambiguë. Le contexte ou des mots supplémentaires sont alors utilisés afin de déterminer le véritable sens.

⁹Le même sens peut être exprimé par différents mots, il y a alors émergence de synonymes.

¹⁰Il n'y a jamais une cohérence totale des agents à cause de leurs différents niveaux d'acquisition. Néanmoins, on observe un succès quasi total de la communication après un nombre suffisant de jeux du langage, car le langage est devenu suffisamment compatible avec tous les contextes d'utilisation.

Nous garderons la représentation binaire pour définir les systèmes de classeurs. Le nombre d'agents est fixé à deux pour les premières expérimentations afin que le modèle soit minimal. Les agents n'ont pour but que de communiquer. Chaque agent dispose de capteurs qui reflètent ce qu'il perçoit. Ces capteurs constituent la partie génétique du système dont ils disposent à leur création mais qu'ils peuvent faire évoluer de manière phylogénique. Le but de chaque agent est donc de "deviner" ce que l'autre agent perçoit. L'intérêt pour l'agent est d'augmenter sa propre perception à l'aide de celle de son voisin pour résoudre ses problèmes. Nous revisitons donc les *language games*.

6.2.1 Structure d'un agent

Un agent A_1 exprime son environnement local en utilisant un mot m_1 qu'il place dans une variable d'environnement globale qui simule un médium de communication parfait. L'agent A_2 lit le mot m_1 depuis la variable d'environnement globale et doit apprendre le sens de ce mot. Comprendre le sens du mot m_1 revient à deviner correctement l'environnement local de A_1 . Chaque agent est symbolisé par un classeur composé de chaîne de 8 bits de longueur. Seule la première partie de la partie condition accepte un alphabet ternaire $\{0, 1, \#\}$. Les autres parties du classeur sont composées de bits définis sur un alphabet binaire $\{0, 1\}$.

La figure 6.1 représente la structure d'un classeur correspondant à l'agent A_1 .



FIG. 6.1 – Structure de l'agent A_1 .

La partie condition se compose de deux parties distinctes de deux bits chacune. La première partie sert à lire le mot qui a été placé dans la variable d'environnement globale. Cette partie est dédiée à la communication. La seconde partie de la partie condition est constituée de l'environnement local perçu par l'agent. Cette seconde partie constitue le lien avec la phylogénèse de l'agent et le but du système communicant est d'apprendre, de manière ontogénique, à correctement identifier cette partie de la règle. Pour ce faire la partie action des classeurs communicants est également constituée de deux parties distinctes de deux bits chacune. La première partie représente le mot qui sera posté dans la variable d'environnement globale. La seconde sert quant à elle à vérifier si l'environnement local de l'autre agent a été correctement supposé.

Voyons maintenant le fonctionnement d'une communication réussie avec la figure 6.2 :

1. L'agent A_1 choisit aléatoirement une règle à activer. Celle-ci lui permet de poster dans la variable d'environnement globale le mot (11), correspondant à son environnement local (01).
2. L'agent A_2 trouve une règle à activer à partir du mot (11) trouvé dans la variable d'environnement globale. La partie environnement local, i.e. la seconde partie de la partie condition, de la

règle choisie par A_2 n'intervient pas dans la sélection de celle-ci.

3. L'agent A_2 utilise alors le second champ de sa partie action pour vérifier qu'il ne s'est pas trompé sur l'environnement local de A_1 : est-ce que l'environnement local de A_1 supposé par A_2 (01) correspond bien à l'environnement local de A_1 (01) ?

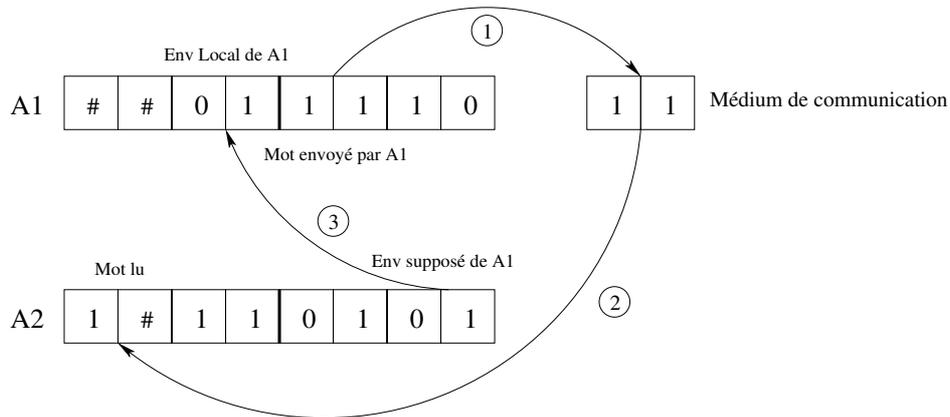


FIG. 6.2 – Exemple de communication réussie.

Cet exemple permet de déduire que dans le lexique de A_2 , le mot 11 provenant de A_1 signifie que l'agent A_1 perçoit 01 comme environnement local : “Quand A_1 dit 11 c'est qu'il perçoit 01.”

Pour que la formation du lexique soit cohérente, nous avons choisi de “donner la parole” aux agents à tour de rôle. Les agents sont strictement identiques toutefois nous supposerons que leur rôle est différent afin de justifier qu'ils communiquent au lieu d'échanger leur savoir. Le but de ce chapitre étant de montrer les propriétés qui émergent lors de la formation d'un langage avec des agents aussi simples que possibles, nous supposerons que les agents sont hétérogènes.

Nous venons de voir comment implémenter les *language games* au niveau des systèmes de classeurs de façon minimale, il s'agit maintenant de trouver une mesure qui reflète bien le succès de la communication.

6.2.2 Mesurer la Communication

Comme nous l'avons vu au chapitre 2, la mesure d'entropie proposée par [MacLennan, 1991, MacLennan et Burghardt, 1994, MacLennan, 1999] a une faiblesse. Dans sa conception originelle, et comme l'explique [Atlan, 1992], la mesure d'entropie sert à quantifier le bruit dans une trame de communication. Il s'agit donc de mesurer si une information transmise, l'a été correctement ou non à travers un médium de communication. Prise telle quelle, cette mesure est unidimensionnelle, tandis que mesurer le bruit qui s'établit dans la formation d'un lexique est une mesure bi-dimensionnelle d'où la faiblesse de la mesure reprise par MacLennan. D'autre part [Noble, 1998]¹¹ a montré en reprenant

¹¹Pages 79 à 81.

les expérimentations de [MacLennan et Burghardt, 1994] que les résultats présentés dans leur article ne reflète pas parfaitement le comportement moyen des simorgs communicants (cf. § 2.3.1). Afin d'illustrer notre propos, regardons la valeur d'entropie de la matrice proposée dans la table 6.1 :

$$H = -\sum_{i,j} p_{ij} \times \log_2(p_{ij}) = -8 \times \left(\frac{1}{8} \times \log_2\left(\frac{1}{8}\right)\right) = \frac{8}{8} \times 3 \times \log_2(2) = 3.0 = H_{\text{Ideale}}$$

Il est flagrant que cette matrice n'est pas une matrice idéale mais reflète plutôt une matrice remplie de synonymes.

		Situation $\lambda (i)$								
		0	1	2	3	4	5	6	7	
S y m b o l e $\gamma(j)$	S	0	1	0	0	0	0	0	0	0
	y	1	1	0	0	0	0	0	0	0
	m	2	1	0	0	0	0	0	0	0
	b	3	1	0	0	0	0	0	0	0
	o	4	1	0	0	0	0	0	0	0
	l	5	1	0	0	0	0	0	0	0
	e	6	1	0	0	0	0	0	0	0
	$\gamma(j)$	7	1	0	0	0	0	0	0	0

TAB. 6.1 – Une fausse matrice idéale.

Nous avons reformulé le calcul de l'entropie afin qu'il prenne en charge l'aspect bi-dimensionnel d'une matrice lexicale. Le calcul principal reste celui de [Shannon et Weaver, 1949] :

$$H = -\sum_{i,j} p_{ij} \times \log_2(p_{ij})$$

Le changement intervient au niveau du calcul de p_{ij} :

$$p_{ij} = \frac{M_{ij}}{\sum_k M_{kj} + \sum_l M_{il} - M_{ij}}$$

Avec M_{ij} qui représente le nombre de succès en communication pour le mot i ayant le sens j . Le nouveau p_{ij} accentue le poids de chaque paire par rapport à la ligne et à la colonne où il se trouve contrairement à l'ancienne mesure qui calculait le poids par rapport à l'ensemble de la matrice. Nous avons désormais pour une matrice de taille $n \times n$:

$$H_{\text{uniforme}}^* = -n^2 \times \frac{1}{(2 \times n) - 1} \log_2\left(\frac{1}{(2 \times n) - 1}\right) = \frac{n^2}{(2 \times n) - 1} \log_2((2 \times n) - 1)$$

$$H_{\text{ideale}}^* = -n \times \frac{1}{1+1-1} \log_2\left(\frac{1}{1+1-1}\right) - (n^2 - n) \times 0 \times \log_2(0) = -n \times \log_2(1) = 0.0$$

Par exemple, avec une matrice 8×8 :

$$H_{\text{uniforme}}^* = \frac{64}{15} \times \log_2(15) \simeq 16.67$$

Enfin, avec la fausse matrice idéale de la table 6.1, nous obtenons :

$$H^* = -8 \times \frac{1}{1+8-1} \times \log_2\left(\frac{1}{1+8-1}\right) = 3 \times \log_2(2) = 3.0 \neq H_{\text{ideale}}^*$$

Afin de pouvoir comparer notre mesure avec celle de [MacLennan et Burghardt, 1994], nous avons établi une mesure qui calcule le taux de succès de la communication en fonction de H , ancienne ou nouvelle mesure :

$$\text{Succes_Communication} = \frac{H_{\text{uniforme}} - H}{H_{\text{uniforme}} - H_{\text{ideale}}} \times 100$$

A l'aide du *Succes_Communication*, nous mesurons la distance du H obtenu par expérimentation, à H_{ideale} dans l'intervalle $[H_{\text{ideale}}, H_{\text{uniforme}}]$. Ce taux de succès montre l'état d'avancement de la

formation d'un lexique dans la population d'agents. Sans contester les résultats obtenus par MacLennan, nous obtenons avec la nouvelle mesure pour les différentes expérimentations les résultats qui se trouvent dans la table 6.2.

Expérimentation	<i>Succes_Communication</i>	<i>Succes_Communication*</i>
Fausse matrice idéale (table 6.1)	100%	82,00%
Ni comm. ni apprentissage (table 2.4)	8,50%	10,50%
Comm. sans apprentissage (table 2.5)	51,25%	48,89%
Comm. et apprentissage (table 2.6)	63,25%	50,75%

TAB. 6.2 – Comparaison des mesures.

Nous constatons qu'il y a des écarts significatifs dans les résultats. Nous remarquons également que le succès de communication est en progrès dans les deux cas. [Noble et Cliff, 1996] le soulignent parfaitement, les simorgs ont un comportement parfois aléatoire qui est dû principalement à la façon dont les simorgs communiquent : c'est à dire dans l'ordre où ils ont été créés et avec le simorg qui se trouve après eux. Cette remarque souligne que l'on ne peut pas exploiter pleinement l'éthologie synthétique dans de telles circonstances. L'idée reste valide mais simplement mal appliquée, ce qui empêche toute généralisation des observations effectuées dans le monde réel. La mesure que nous proposons, montre bien que la communication n'est pas parfaitement établie entre les simorgs. Le constat fait par [Noble,1998]¹² est la principale raison de cet échec. Les résultats obtenus avec la fausse matrice idéale indiquent que l'ancien calcul ne permet pas de mesurer le stade d'évolution de la formation d'un lexique. La nouvelle mesure indique qu'il y a bien communication puisque le système identifie clairement 8 sens avec le premier mot toutefois on ne peut parler de lexique puisque chacun des sens n'est pas clairement distingué par un mot unique, i.e. le lexique n'est pas encore formé.

6.2.3 Conclusion

Nous venons de définir un modèle minimal de communication qui se compose de deux systèmes de classeurs qui représentent chacun des agents et qui communiquent en jouant au *language games* et plus particulièrement au *naming games* puisque le but du jeu n'est pas d'apprendre à parler mais d'apprendre à nommer des "environnements locaux" contenus dans les classeurs des agents communicants. Nous avons défini la structure que doit avoir chaque classeur pour communiquer de manière minimale. Les agents hétérogènes communiqueront à tour de rôle et nous mesureront leur performance à l'aide de la nouvelle mesure *Succes_Communication* qui reflète le taux de formation du lexique des agents à l'aide de la nouvelle mesure d'entropie. Nous allons voir maintenant quelques exemples d'application du modèle minimal, aussi bien au niveau de l'éthologie synthétique que de robots simulés.

¹²Pages 81 à 87.

6.3 Application

Cette partie présente trois implémentations du modèle minimal de communication : une première expérimentation basée sur des systèmes de classeurs de type Pittsburgh simplifié (cf. § 4.3.3), une seconde basée sur le XCS (cf. § 3.2.4) afin de vérifier que notre approche peut s'appliquer à d'autres systèmes de classeurs et enfin nous avons effectué une expérimentation en utilisant des robots simulés afin d'étudier la possibilité d'ancrage du modèle dans la réalité.

6.3.1 Systèmes de classeurs de type Pittsburgh simplifié

A partir du modèle minimal défini précédemment, nous avons fait une première expérimentation pour laquelle chaque agent est composé d'un système de classeurs dont les caractéristiques se trouvent dans les tables 6.4 et 6.6.

Paramètre	Réglage
Nombre d'agents	2
Nombre d'individus (<i>NombreIndividus</i>)	14
Nombre de règles par individu (<i>NombreRegles</i>)	4
Nombre d'essais entre générations	10

TAB. 6.4 – Paramètres du système multi-agents.

Paramètre	Réglage
$P_{\#}$	0%
$P_{\text{Croisement}}$	60%
P_{Mutation}	0,05%
<i>Elitisme</i>	$\frac{2}{14} \times 100 = 14,29\%$
Mécanisme de Sélection	Roulette Wheel
Sélection des règles (cf. § 4.3.3)	Spécifique
Croisement	Monopoint
Nombre de Générations	1000

TAB. 6.6 – Paramètres de l'algorithme génétique.

Chaque agent doit "deviner" l'environnement local de l'autre agent à partir du mot que cet autre agent aura posté. Dans la définition du modèle minimal, nous avons deux bits qui expriment un mot et deux bits pour l'environnement local. Il y a donc quatre mots possibles pour deviner quatre environnements locaux. La mesure d'entropie sera calculée à partir d'une matrice de taille 4×4 . Cette matrice

sera remplie à chaque fois qu'une communication est réussie entre les deux agents en ajoutant 1 à la paire $\{mot, environnement\ local\}$ correspondante. La fitness de l'agent correspond aux nombres de succès de communication qu'il a obtenu durant les essais de communication qui ont lieu entre deux générations. Les agents communiquent à tour de rôle et le classeur qui déclenche la communication est choisi au hasard parmi les quatre dont dispose l'individu en cours d'évaluation. Pour ces expérimentations, nous avons laissé la mutation créer des caractères génériques $\{\#\}$ uniquement dans le premier champ de la partie condition. Nous avons moyenné les résultats sur 100 expérimentations en utilisant différentes racines aléatoires.

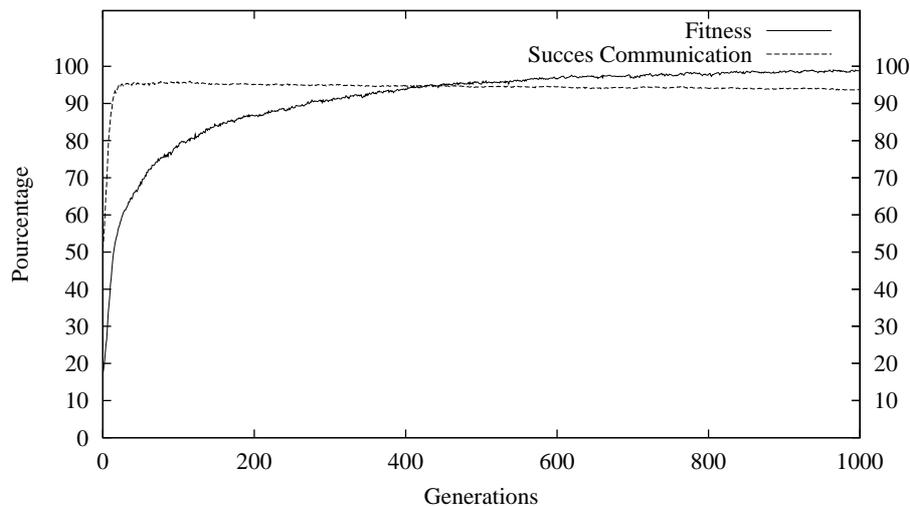


FIG. 6.3 – Résultats Modèle Minimal.

La figure 6.3 montre l'évolution de la valeur de fitness moyenne des agents ainsi que l'évolution du succès de communication au cours des générations. En 20 générations, le taux de succès de communication dépasse 95% puis se stabilise à 93,77% en fin d'évolution. La fitness quant à elle, atteint au final 99% mais demande plus de 300 générations pour dépasser 90%. La progression rapide du succès de communication s'explique par le fait que le remplissage de la matrice dénotationnelle ne se fait que lorsqu'il y a un succès de communication et ne prend pas en compte les échecs qui se reflètent dans la valeur de fitness. Le lexique final se forme dès le début de l'évolution. Par contre, la diffusion de ce lexique au sein des classeurs, et éventuellement l'élargissement de celui-ci, demande plus de temps. Nous avons choisi de regarder deux des individus des deux populations de classeurs à la dernière génération de l'une des expérimentations choisie au hasard. Ces individus sont représentatifs de l'ensemble des 14 individus de chacun des systèmes de classeurs puisque la population s'est uniformisée (cf. chapitre 4). Du fait de l'uniformisation, on peut généraliser ces résultats aux agents eux-mêmes. Chacun de ces "agents" est représenté dans la table 6.7 avec leurs 4 classeurs respectifs.

De ces deux agents, nous avons déduit la table 6.8 qui reflète l'ensemble des communications possibles entre les agents en fonction du classeur choisi pour commencer la communication. Pour

Agent 1				Agent 2			
C_1	01	01 : 01	00	C'_1	10	00 : 01	11
C_2	0*	11 : 00	10	C'_2	*0	10 : 10	11
C_3	*0	01 : 01	10	C'_3	01	10 : 10	01
C_4	0*	11 : 10	11	C'_4	1*	00 : 01	01

TAB. 6.7 – Exemple de deux individus en fin d'expérimentation.

chaque agent, on décompose ses classeurs en mots possibles à émettre, en réponse possible de l'autre agent sur l'environnement local correspondant au mot émis et enfin en environnement local réel de l'agent qui a émis le mot.

	Mot posté	Env. local supposé	Env. local réel
A_1	00	11 (C'_2)	11
	01	01 (C'_3)	01
	10	11 (C'_1 or C'_2) or 01 (C'_4)	11
A_2	01	00 (C_1) or 11 (C_2 or C_4)	00
	10	10 (C_3)	10

TAB. 6.8 – Communications possibles.

Premier constat : l'agent 2 (A_2) connaît le sens des 3 différents mots émis par l'agent 1 (A_1) : 00, 01 et 10. Prenons par exemple le cas où A_1 poste le mot 10 dans la variable d'environnement global, l'agent 2 peut alors activer les classeurs C'_1 , C'_2 ou C'_4 . Les deux derniers classeurs cités auront moins de chances d'être activés du fait de leur faible spécificité. Le classeur C'_1 est alors activé. Ce classeur prédit que l'environnement local de A_1 est 11, ce qui est le cas. Les deux individus représentant les agents sont alors récompensés et la matrice dénotationnelle remplie en conséquence.

Second constat : A_1 connaît quant à lui le sens des 2 mots que peut émettre A_2 : 01 et 10. De la même manière que dans l'exemple précédent, le classeur C_1 sera sélectionné pour sa spécificité et permettra une communication réussie.

Dernier constat : dans l'exemple choisi, l'agent A_1 utilise les mots 00 et 10 pour désigner l'environnement local 11. Il y a donc synonymie entre ces deux mots, c'est un phénomène que nous mesurerons lors de l'extension du modèle.

Nous avons obtenu deux agents capables de communiquer entre eux quel que soit le mot émis. Nous avons toutefois voulu vérifier la robustesse de notre approche en vérifiant qu'un autre système de classeurs était capable de gérer de la même manière la communication avec le modèle minimal [Énée et Esczut, 2001a].

6.3.2 Systèmes de classeurs de type XCS

Les systèmes de classeurs de type XCS permettent de garder telle quelle la structure que nous avons choisi d'implémenter puisqu'ils se composent de classeurs qui peuvent également être des chaînes de bits. La table 6.9 permet de comparer les réglages qui sont communs aux deux systèmes. On constate que les deux systèmes sont très proches excepté dans leur mode d'évaluation (cf. § 3.2.3 et § 3.2.4) : les individus de la version Pittsburgh simplifié du modèle minimal sont évalués 10 fois entre deux générations afin de permettre à tous les classeurs des individus de s'exprimer, contrairement au XCS où ils ne sont évalués qu'une seule fois.

	Pittsburgh	XCS
Nombre d'expérimentations	100	100
Type de population initiale	Aléatoire	Vide
Nombre de générations	1000	1000
Taille Population	14	60
Classeurs par individu	4	1
Croisement	Monopoint	Monopoint
$P_{\text{Croisement}}$	60%	60%
P_{Mutation}	0, 1%	1%
$P_{\#}$	aucun	20%
Élitisme	14, 29%	aucun

TAB. 6.9 – Comparaison des paramètres.

Certains paramètres sont propres aux systèmes de classeurs de type XCS. Nous avons choisi de reprendre les paramètres utilisés par [Butz et Wilson, 2000] que l'on peut retrouver dans la table 6.10.

Initial Prediction	0.01	α	0.1	θ_{del}	20
Initial Error	0.01	β	0.2	θ_{GA}	40
Initial Fitness	0.01	δ	0.1	θ_{sub}	20
		ε_0	0.01	θ_{mma}	2
		γ	0.71		
		ν	5		

TAB. 6.10 – Paramètres XCS.

Les résultats sont présentés par la figure 6.4. On constate que les deux systèmes de classeurs se comportent parfaitement. En moins de 200 générations on atteint un taux satisfaisant de succès de communication qui permet de dire que les lexiques sont en cours de finalisation. Les taux atteignent, à la dernière génération, 93,77% pour le Pittsburgh simplifié et 85,19% pour le XCS. Si l'on poursuit l'expérience, les deux systèmes se rejoignent dans les 1500 générations suivantes.

Lors de cette expérience, nous avons obtenu des résultats qui confirment que l'on peut appliquer

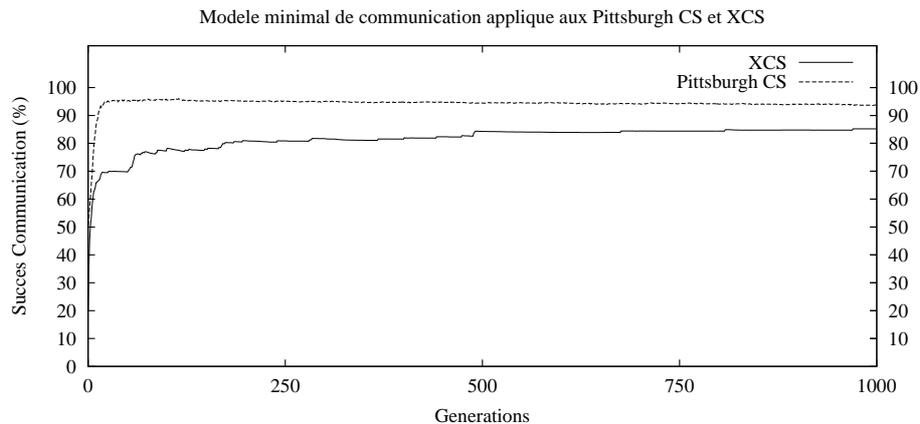


FIG. 6.4 – Modèle minimal appliqué au Pittsburgh CS et au XCS.

le modèle minimal de communication à d'autres systèmes de classeurs. Il semblerait que l'on puisse appliquer le modèle à n'importe quel système apprenant de type réactif avec une entrée et une sortie. Nous nous tournons maintenant vers une forme d'ancrage dans la réalité par la simulation d'un problème réel résolu par le modèle minimal.

6.3.3 Le robot Khepera

[Steels et Kaplan, 1999] affirment :

Although this formal approach has many virtues, it makes a number of simplifying assumptions which are not valid for physically grounded evolving autonomous agents that are part of inhomogeneous populations operating in real world open environments.¹³

The physical environment produces an infinite set of unforeseeable situations, independent of the agent and not delineatable by an outside observer.¹⁴

[Dorigo et Schnepf, 1992] précisent également :

An autonomous agent must possess this adaptive power itself in order to adapt its behaviour to any changes in the environment.¹⁵

Pour cette raison, nous avons voulu appliquer le modèle minimal de communication à un problème de robotique qui permet d'ancrer le système dans la réalité [Kaplan, 1999], même si celle-ci reste simulée.

¹³Même si l'approche formelle a de nombreuses vertus, elle suppose un certain nombre de simplifications qui ne sont pas valides avec des agents réels, évoluant et autonomes qui font partie de population non homogènes qui agissent dans des environnements réels et ouverts.

¹⁴L'environnement physique produit un ensemble infini de situations imprédictibles, indépendantes de l'agent et qui ne peuvent être décrites par un observateur externe.

¹⁵Un agent autonome doit disposer de la capacité à s'adapter pour pouvoir adapter son comportement à tout changement de l'environnement.

Nous allons maintenant tenter d'appliquer le modèle minimal à un problème de robotique, le jeu des positions [Xuan et al., 2001]. Nous avons utilisé pour cela un simulateur de robot Khepera implémenté par [Michel, 1995]. Le Khepera (cf. figure 6.5) est un petit robot de 5 centimètres de diamètre qui dispose de 8 capteurs infrarouges et de deux moteurs avec des roues. Le simulateur représente un carré de 1×1 m dans lequel nous avons fait évoluer les deux robots qui représentent chacun des agents tentant de communiquer.



FIG. 6.5 – Le robot Khepera

Le jeu des positions consiste pour chaque robot à deviner où se trouve l'autre. Dans un effort de simplification du problème, nous avons défini un monde circulaire dont chacun des quatre points cardinaux représentent une position à laquelle peut se trouver un robot. Chaque robot peut se rendre d'une position à une autre en un seul mouvement. Nous avons divisé le cercle en quatre afin de pouvoir représenter les positions à l'aide de deux bits. Un robot tente de deviner l'endroit où se trouve l'autre robot à partir du mot de deux bits qu'il reçoit par l'intermédiaire de la variable d'environnement global. Ce robot se déplace alors à la nouvelle position et sera récompensé en fonction de la distance à laquelle il se trouve de l'autre robot.

Dans cet expérience, la principale difficulté est que les deux robots ne peuvent activer une règle que si leur propre environnement local correspond à la position où il se trouve. Nous allons voir un exemple de deux robots K_1 et K_2 qui sont aux positions représentées par la figure 6.6.

La table 6.11 contient les deux individus en cours d'évaluation, correspondant à K_1 et K_2 .

	K 1					K 2			
C_1	11	00	: 01	00	C'_1	01	10	: 01	00
C_2	11	01	: 10	00	C'_2	10	00	: 11	01
C_3	10	01	: 10	00	C'_3	10	10	: 11	11
C_4	01	10	: 11	10	C'_4	11	01	: 10	10

TAB. 6.11 – Robots en cours d'évaluation.

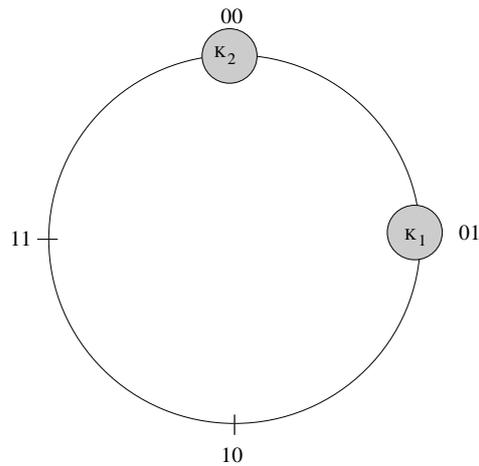


FIG. 6.6 – Exemple de monde simulé des Khépéras.

Supposons que ce soit au tour de K_1 de poster un message, du fait de sa position, il ne peut choisir qu'une règle ayant comme environnement local 01. C'est le cas du classeur C_2 qu'il va pouvoir déclencher. Le mot 10 est donc posté dans la variable d'environnement globale. K_2 lit le mot posté et recherche une règle qui va à la fois correspondre au mot 10 mais également à sa position actuelle, c'est à dire 00. Pour K_2 , seul le classeur C'_2 répond à ces critères. Il suppose alors que le robot K_1 est dans la position 01 d'après le deuxième champ de sa partie action. Il se rend alors en 01 et poste le mot 11. K_2 est alors récompensé puisqu'il a prédit la position de K_1 . C'est alors au tour de K_1 de deviner la position de K_2 . À partir du mot posté par K_2 (11) et de sa position actuelle (01), K_1 ne peut activer que le classeur C_2 . Il se rend alors en 00 où K_2 ne se trouve plus. La récompense de K_1 sera dépendante de la distance à laquelle se trouve le robot qui bouge de celui qu'il doit rejoindre :

$$Distance = Position(RobotEcoute) - Position(RobotParle)$$

$$Distance_{Max} = 2$$

$$Recompense = \frac{Distance_{Max} - Distance}{Distance_{Max}}$$

La table 6.12 contient les réglages qui sont différents de ceux du modèle minimal.

Paramètre	Réglage
Nombre d'expérimentations	20
Nombre de générations	200

TAB. 6.12 – Paramètres Khépéra.

Les résultats se trouvent sur la figure 6.7. Nous obtenons un taux final de 83,59% de succès de la communication, ce qui indique que le lexique est en cours de finalisation puisque le taux de 80% est atteint en seulement 14 générations. Ce résultat reste en dessous de ce que nous obtenions avec le modèle minimal simple. Ceci est tout à fait normal puisque l'ancrage dans la réalité simulée multiplie par 4 le nombre de règles nécessaires pour chaque mot posté or le nombre de classeurs par

individus n'a pas évolué. Par conséquent le résultat est plus qu'encourageant et permet de confirmer que les robots ont appris à se comprendre par l'intermédiaire d'un lexique [Énée et Esczut, 2001b]. L'observation du simulateur permet de constater deux comportements de la part des robots. Soit ils oscillent entre deux positions, soit ils se contentent de rester à la même place, c'est à dire le plus souvent sur la position 00. Là encore, ce comportement s'explique parfaitement du fait du faible nombre de règles par individu.

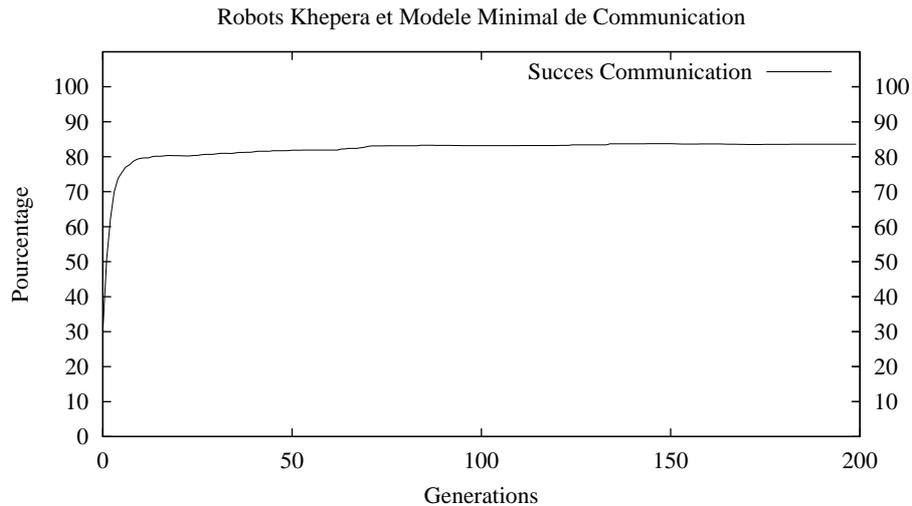


FIG. 6.7 – Résultats Khépéras et modèle minimal.

6.3.4 Conclusion

Nous avons fait un tour d'horizon d'un modèle de communication que nous avons défini le plus simplement possible. Il se compose de deux agents interagissant pour former un lexique leur permettant de comprendre ce que perçoit l'autre agent. Le modèle théorique a parfaitement fonctionné avec des systèmes de classeurs de type Pittsburgh simplifié. Nous avons ensuite voulu vérifier la robustesse de notre approche en l'appliquant avec un autre système de classeurs, le XCS. Là encore le modèle minimal de communication s'est avéré parfaitement adaptable et nous a donné entière satisfaction. Enfin, nous avons voulu ancrer le modèle à la réalité en l'appliquant à un problème concret de robotique, le jeu des positions. Les robots Khépéra simulés qui nous ont servi de support ont appris à communiquer à l'aide du modèle minimal malgré un surcroît de difficultés liées à la prise en compte de l'environnement local dans le choix des règles à activer. Le modèle minimal a donc fait ses preuves [Énée et Esczut, 2002], nous allons voir maintenant comment nous pouvons étendre celui-ci et comment nous allons prendre en compte les problèmes de confusion qui peuvent apparaître au cours de la formation d'un lexique.

6.4 Extension du modèle minimal

Le modèle minimal est intéressant d'un point de vue théorique puisqu'il donne les fondations pour établir une communication entre deux agents appartenant à une population hétérogène. Nous avons donc souhaité étendre de diverses manières l'étude du modèle de communication que nous avons défini. Comme nous l'avons déjà constaté, communiquer coûte aux agents et la communication n'apparaît que si elle permet à différentes catégories d'agents spécialisés d'apparaître dans l'ensemble de la communauté. Nous avons voulu étudier la communication lorsqu'on augmente le nombre des agents et lorsqu'on augmente la taille de la matrice dénotationnelle, i.e. du lexique [Steels et McIntyre, 1999]. Par conséquent, nous avons dû définir une mesure de l'homonymie et de la synonymie comme nous le suggérait les tests sur le modèle minimal. Enfin, nous avons étudié l'évolution de la communication lorsque deux groupes d'agents qui avaient appris séparément à communiquer se rencontrent de diverses manières.

6.4.1 Vers la complexité ...

Nous avons étudié l'influence que peut avoir la complexification de la communication sur celle-ci. Nous avons pour cela, imposé que les agents qui communiquent dans un groupe d'agents, le fassent de manière totale. C'est à dire que, pour apprendre, chaque agent se doit de communiquer avec tous les autres agents de la communauté à laquelle il appartient. Ce critère permet, a priori, de diffuser plus rapidement les choix lexicaux fait par les agents. Cette communication a lieu à tour de rôle. Chaque agent communique par paire avec un autre puis un autre et ainsi de suite jusqu'à ce qu'il ait communiqué avec tous les autres. Ainsi, lors de chaque tentative de communication, les agents utilisent le modèle minimal de communication. Chaque agent dispose d'une matrice dénotationnelle pour chacun des autres agents avec lequel il communique.

Le nombre de mots et d'environnements locaux, i.e. la taille du lexique, des agents est représenté par le nombre de bits n qui les compose. Le nombre de mots m est par conséquent :

$$m = 2^n$$

Nous avons fait varier le nombre d'agents de 2 à 10 et le nombre de bits de 2 à 5, i.e. le nombre de mots et d'environnements locaux de 4 à 32. Lors de toutes nos expériences, le nombre de mots et le nombre d'environnements locaux possibles augmentent de concert. La table 6.13 donne les paramètres de l'expérimentation et la table 6.14 rappelle l'ensemble des paramètres de l'algorithme génétique. Les résultats ont été moyennés sur 100 expérimentations avec différentes racines aléatoires. Nous avons fixé le nombre d'individus et de règles pour l'ensemble des expérimentations, nous verrons que cela peut avoir une influence sur le comportement du système comme nous avons déjà pu l'observer au chapitre 4.

La figure 6.8 montre les résultats obtenus avec 4 mots et 4 environnements locaux pour chaque agent, i.e. une taille de lexique de 4.

Après un établissement rapide de la communication avec un maximum de 92% à la génération 13,

Paramètre	Réglage
Nombre d'agents	2 à 10
Nombre de bits par mot et env. local	2 à 5
Nombre d'individus (<i>NombreIndividus</i>)	10
Nombre de règles par individu (<i>NombreRgles</i>)	12
Nombre d'essais entre générations	20

TAB. 6.13 – Paramètres du système multi-agents.

Paramètre	Réglage
$P_{\#}$	0%
$P_{\text{Croisement}}$	70%
P_{Mutation}	0,1%
<i>Elitisme</i>	$\frac{2}{10} \times 100 = 20\%$
Mécanisme de Sélection	Roulette Wheel
Sélection des règles (cf. § 4.3.3)	Spécifique
Croisement	Monopoint
Nombre de Générations	5000

TAB. 6.14 – Paramètres de l'algorithme génétique.

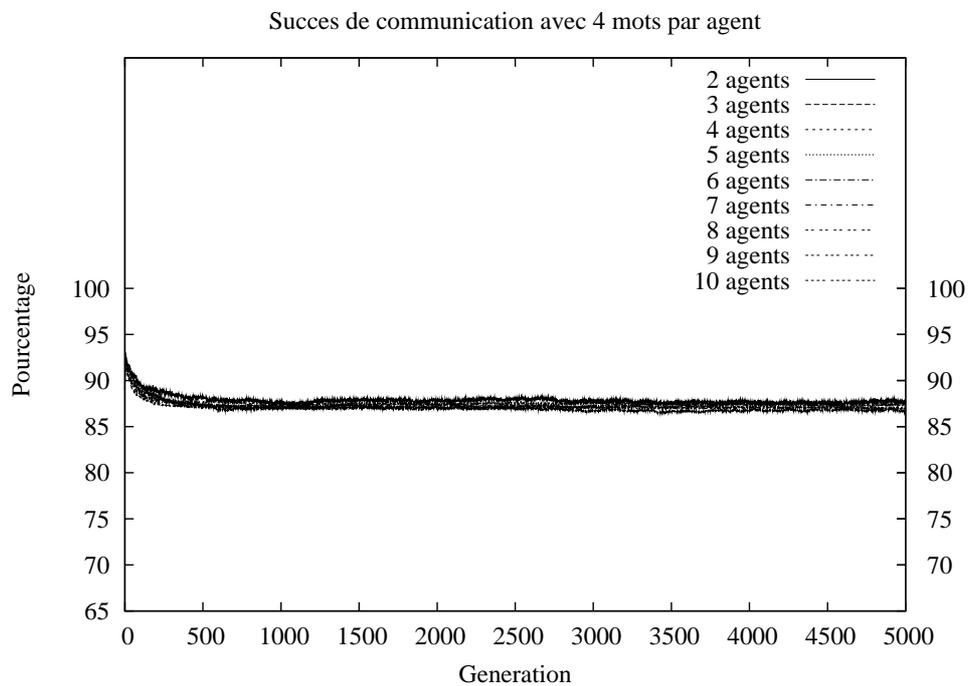


FIG. 6.8 – Résultats avec 4 mots et 4 environnements locaux.

le système se stabilise autour de 87% à la fin des expérimentations quel que soit le nombre d'agents qui communiquent. Il semble donc qu'un lexique se soit formé et se maintienne au cours du temps. Le nombre d'agents ne semble pas influencer le succès de la communication, ce résultat ne peut encore être expliqué, il nous faut poursuivre les expérimentations en regardant le comportement du système lorsque le nombre de mots possibles évolue de 4 à 8.

La figure 6.9 montre les résultats avec un lexique de taille 8.

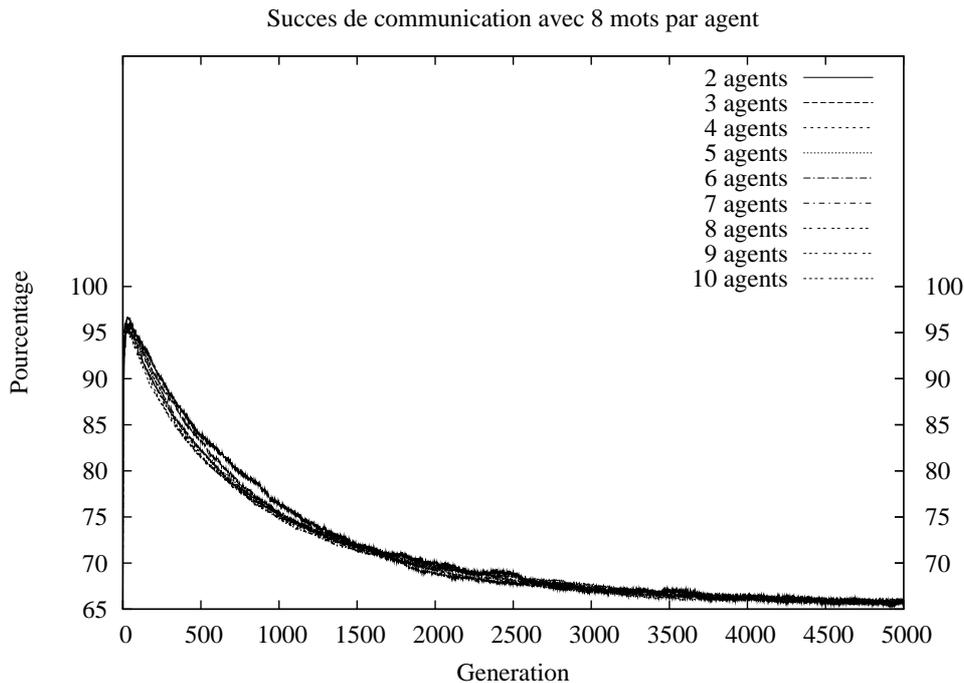


FIG. 6.9 – Résultats avec 8 mots et 8 environnements locaux.

La communication atteint rapidement un maximum d'un peu plus de 96% à la génération 44. Nous observons ensuite un effondrement progressif du succès de la communication quel que soit le nombre d'agents. Le taux final de communication dépasse à peine 65%. Si l'on compare ce nombre avec les résultats obtenus par [MacLennan et Burghardt, 1994], nous savons qu'il y a un lexique dans ces populations d'agents mais également qu'une grande confusion règne. Nous allons devoir comprendre d'où provient cette confusion en mesurant d'autres caractéristiques des lexiques en cours de formation. Là encore le nombre d'agents ne semble avoir aucune influence sur les résultats observés. Nous allons regarder maintenant si le succès de communication continue à s'effondrer lorsque le nombre de mots et d'environnements locaux passe de 8 à 16.

La figure 6.10 contient les résultats avec 16 mots utilisables par les agents.

On observe de nouveau un pic du succès de la communication avec 96,82% à la génération 108. Avec l'augmentation de la taille du lexique, le pic du maximum de succès de la communication se produit de plus en plus tard et les courbes semblent se lisser. Le taux final avoisine les 75,70%. Ce

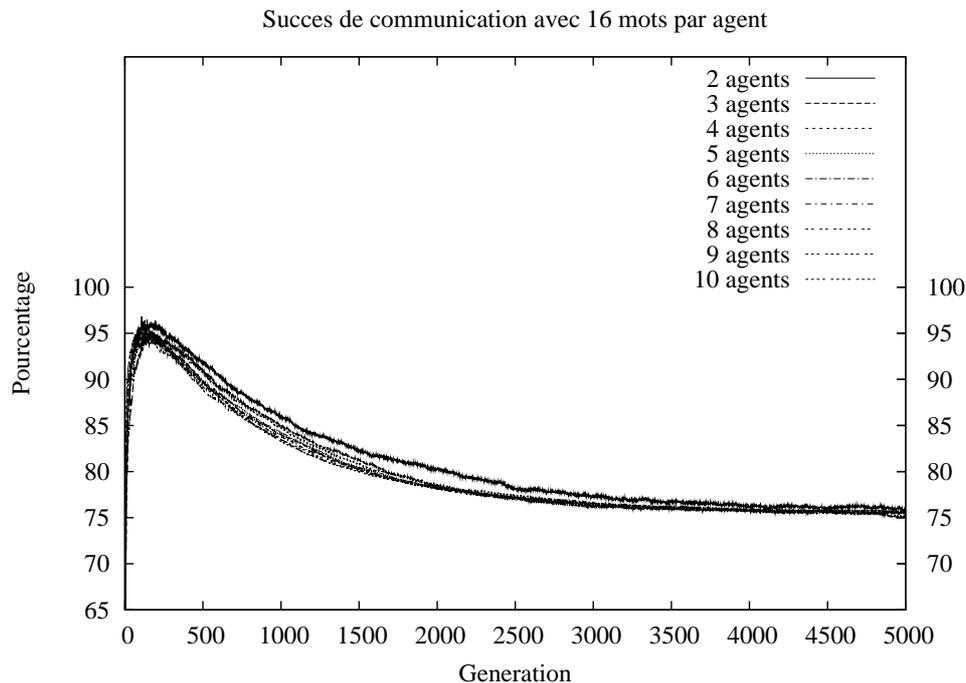


FIG. 6.10 – Résultats avec 16 mots et 16 environnements locaux.

résultat est bien meilleur que pour les 8 mots. Il semble donc que la complexification des agents ne soit pas corrélée à une baisse du succès de la communication. Pour autant, nous ne pouvons encore dire pour quelles raisons l'expérience avec 8 mots était moins bonne. Si l'on regarde de plus près la courbe pour 2 agents, on s'aperçoit que celle-ci reste légèrement au dessus des autres. La dynamique du système semble progressivement se stabiliser, c'est pourquoi nous avons augmenté encore une fois le nombre de mots en le passant de 16 à 32.

Les résultats avec 32 mots sont présentés dans la figure 6.11.

Le comportement du système est complètement différent des autres. Il y a cette fois-ci une différence significative entre la courbe à 2 agents et celle à 10 agents. En effet, on observe que pour la courbe à 2 agents, celle qui est la plus basse, passe par un maximum de 95,91% à la génération 675 puis redescend progressivement aux alentours de 88,25%. Pour la courbe à 10 agents, le maximum est atteint à la génération 406 avec 99,29% puis le succès de communication se stabilise autour de 98,53%. L'augmentation du nombre d'agents détermine directement le taux de stabilisation du succès de communication. Plus ce nombre est grand, plus le succès de communication est élevé. Le gain en succès de communication suit une courbe logarithmique dépendante du nombre d'agents au delà de 2. Il semble que les paramètres choisis soient responsables de ce résultat. En effet, nous arrivons aux limites de ce que peuvent gérer 10 individus munis de 12 classeurs [Wagner, 1999].

On peut également faire un deuxième constat lorsque le nombre de mots augmente, on crée une confusion dans la communication : il y a trop et en même temps pas assez de mots et de sens pour

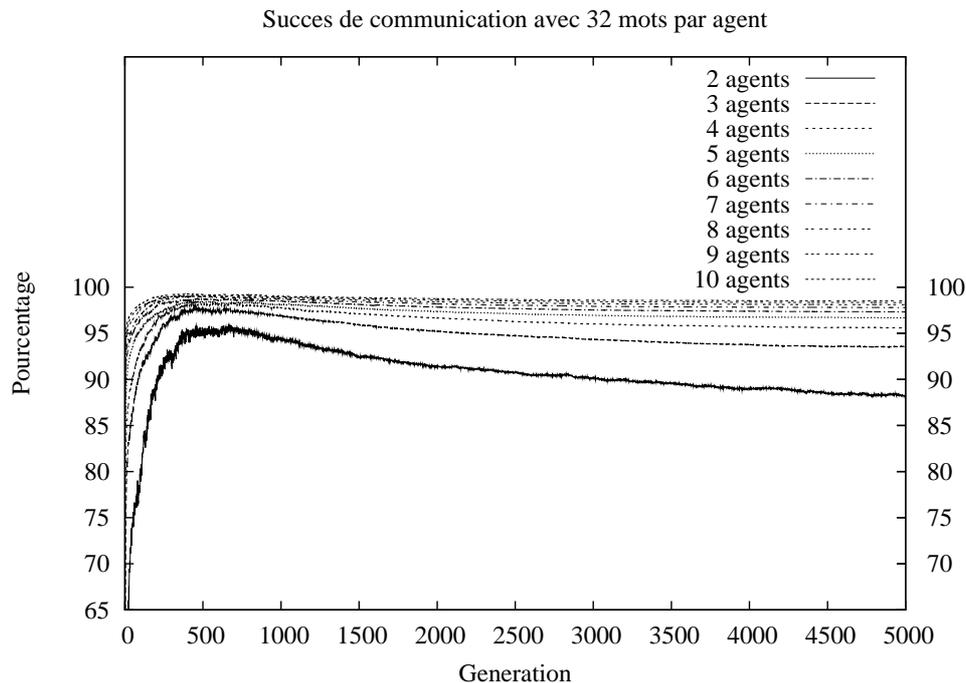


FIG. 6.11 – Résultats avec 32 mots et 32 environnements locaux.

permettre à un lexique clair de se former [Kaplan, 2000]. Lorsqu'on utilise le modèle minimal aucune confusion ne s'installe, celle-ci apparaît lorsqu'on passe à 8 mots. Le fait qu'il y ait 12 règles par individu peut justifier que la confusion s'installe puisqu'à l'aide de 12 règles on peut distinguer plus de 8 mots et plus de 8 sens sans pour autant avoir une relation bijective. Tester les 64 possibilités est quasiment impossible au niveau individuel à cause de l'uniformisation de ceux-ci, bien que les 10 individus par agent de départ permettent théoriquement de le faire. L'augmentation du nombre de mots transforme cette relation bijective par l'appauvrissement du facteur nombre de règles par rapport aux nombres de combinaisons possibles. En effet, on peut difficilement distinguer 16 mots avec 12 règles, par conséquent, le système risque d'avoir tendance à utiliser les mots de manière unique. Ce phénomène s'amplifie avec 32 mots. A l'inverse lorsqu'il n'y a que 4 mots et 4 sens distinguables, les 12 règles permettent pratiquement d'évaluer l'ensemble des 16 combinaisons possibles et d'éliminer progressivement celles qui sont inutiles.

Nous allons vérifier cette hypothèse à l'aide d'expérimentations complémentaires mesurant l'état de confusion dans les lexiques en cours de formation.

6.4.2 Analyse du lexique

Nous avons défini un modèle de communication pour lequel la fonction de récompense favorise le développement de la communication entre les agents vers la formation de lexiques idéaux. Pourtant, même dans ce cas de figure idéale, et comme l'a constaté [Edwin De Jong, 2000] :

Inspection revealed that words did not always uniquely identify a meaning, but were sometimes associated with several meanings.¹⁶

La confusion dans la formation des lexiques s'exprime par la tendance des agents à utiliser plusieurs mots pour désigner une chose (synonymie) ou au contraire à n'utiliser qu'un mot pour désigner plusieurs choses (homonymie). Nous allons voir l'influence réelle sur la formation des lexiques de l'augmentation du nombre de mots, en terme d'homonymie et de synonymie.

Homonymie L'homonymie se traduit par l'utilisation d'un mot pour désigner plusieurs sens différents. Au niveau de la matrice dénotationnelle, nous pouvons calculer un pourcentage d'homonymie dans un lexique en modifiant légèrement la mesure d'entropie et plus particulièrement le calcul de p_{ij} qui, pour mémoire, donne le poids relatif du mot i et du sens j dans la matrice :

$$p_{ij} = \frac{M_{ij}}{\sum_k M_{ik}}$$

On obtient donc une pondération de M_{ij} par rapport à l'ensemble des autres sens possibles pour le mot i . On peut alors calculer l'entropie de l'homonymie :

$$H_{Homonymie} = -\sum_{ij} p_{ij} \times \log_2(p_{ij})$$

Afin de normaliser le calcul comme pour le succès de communication, nous calculons pour une matrice dénotationnelle de taille $n \times n$:

$$H_{Homonymie_{Uniforme}} = -n^2 \times \frac{1}{n} \times \log_2\left(\frac{1}{n}\right) = n \times \log_2(n)$$

$$H_{Homonymie_{Ideale}} = -n \times 1 \times \log_2(1) = 0.0$$

Le taux d'homonymie s'exprime alors comme une distance à l'idéale dans l'intervalle représenté par $[H_{Homonymie_{Ideale}}, H_{Homonymie_{Uniforme}}]$:

$$Homonymie = \frac{H_{Homonymie}}{H_{Homonymie_{Uniforme}} - H_{Homonymie_{Ideale}}} \times 100$$

Synonymie La synonymie signifie que plusieurs mots ont le même sens. De la même manière que pour l'homonymie, nous pouvons calculer un pourcentage de synonymie dans un lexique en modifiant la mesure d'entropie par le calcul de p_{ij} :

$$p_{ij} = \frac{M_{ij}}{\sum_j M_{ij}}$$

Le calcul de p_{ij} permet de pondérer chaque sens par rapports aux mots qui le désigne. Le calcul de l'entropie pour la synonymie reste le même :

$$H_{Synonymie} = -\sum_{ij} p_{ij} \times \log_2(p_{ij})$$

Dans le même soucis de normalisation des calculs, nous calculons pour une matrice dénotationnelle de taille $n \times n$:

$$H_{Synonymie_{Uniforme}} = -n^2 \times \frac{1}{n} \times \log_2\left(\frac{1}{n}\right) = n \times \log_2(n)$$

$$H_{Synonymie_{Ideale}} = -n \times 1 \times \log_2(1) = 0.0$$

Le taux de synonymie se calcule alors de la même manière que le taux d'homonymie, comme une distance à l'idéale dans l'intervalle $[H_{Synonymie_{Ideale}}, H_{Synonymie_{Uniforme}}]$:

¹⁶L'inspection (du lexique) révèle que les mots n'identifient pas toujours un sens de manière unique, mais parfois sont associés avec plusieurs sens.

$$\text{Synonymie} = \frac{H_{\text{Synonymie}}}{H_{\text{SynonymieUniforme}} - H_{\text{SynonymieIdeale}}} \times 100$$

Étude A l'aide de ces deux mesures, nous allons revisiter les expériences faites dans la section 6.4.1. Sans changer les paramètres des expérimentations, nous avons mesuré l'évolution de l'homonymie et de la synonymie pour les différents nombres de mots proposés avec pour chaque représentation, les résultats de 2 à 10 agents puisque nous voulons également vérifier les hypothèses que nous avons faites sur l'influence des paramètres.

La figure 6.12 contient les résultats d'homonymie et de synonymie avec 4 mots et 4 sens possibles.

Nous avons obtenu un succès de communication qui passait par un pic à 92% puis une stabilisation autour de 87%. Au moment où la communication réussit à 92%, l'homonymie est à 0% et la synonymie atteint 6,41%. C'est alors que le taux de synonymie ne cesse d'augmenter pour atteindre 9,90% en fin d'expérimentation. La quantité de classeurs disponibles semble permettre aux agents d'utiliser de manière efficace bien qu'inutile un surcroît de mots pour désigner les 4 environnements locaux disponibles. Toutefois l'homonymie à 0% combinée à ce taux de synonymie révèle que les agents ne désignent pas l'ensemble des environnements locaux possibles. En effet, si tous les environnements locaux étaient désignés dans la matrice dénotationnelle, nous aurions soit un taux de synonymie et d'homonymie à 0% pour le cas de la communication parfaite, soit un taux supérieur pour les deux du fait du calcul de p_{ij} . Nous pouvons donc déduire qu'en moyenne, rarement la totalité des environnements locaux sont reconnus, de plus ils ne sont pas systématiquement désignés de manière unique. Regardons maintenant l'évolution de l'homonymie et de la synonymie lorsqu'on passe à un lexique de taille 8.

Nous pouvons voir ces résultats sur la figure 6.13.

Lorsque le succès de communication atteint un maximum de 96% à la génération 44, l'homonymie est de 1,53% et la synonymie atteint 2,14%. Ces taux montrent l'établissement d'un lexique entre les agents sans pour autant garantir que l'ensemble des agents a réussi à communiquer. Nous avons relevé la valeur de fitness moyenne des agents à cet instant qui reflète le nombre de fois qu'ils ont réussi à communiquer en moyenne : ce taux est de 35,25%. Si l'on regarde l'évolution de l'homonymie et de la synonymie par la suite, on observe que le taux d'homonymie dépasse légèrement les 25% et que le taux de synonymie se situe aux alentours de 22%. Dans le même temps, la valeur moyenne des agents atteint 92,58% et le taux de formation lexicale est stabilisé à un peu plus de 65%. Ces résultats semblent indiquer que les agents ont créé deux lexiques qu'ils utilisent alternativement, créant la confusion dans la formation du lexique. Le nombre de règles par individu semble directement responsable de ce phénomène. Il y a en effet 12 règles pour exprimer 8 sens, et si l'on suppose le système capable d'organiser deux lexiques, on s'aperçoit que l'intersection des deux lexiques représente 4 règles sur 12 soit $\frac{1}{3}$ de confusion que l'on retrouve dans les 65% mesuré dans le taux de formation du lexique. L'explication peut alors s'appliquer pour l'expérience à 4 mots dans laquelle le système est capable de créer jusqu'à 3 lexiques distincts sans intersection. Nous allons voir si lors du passage à 16 mots le phénomène diminue puisque les agents pourront alors difficilement avoir plusieurs lexiques.

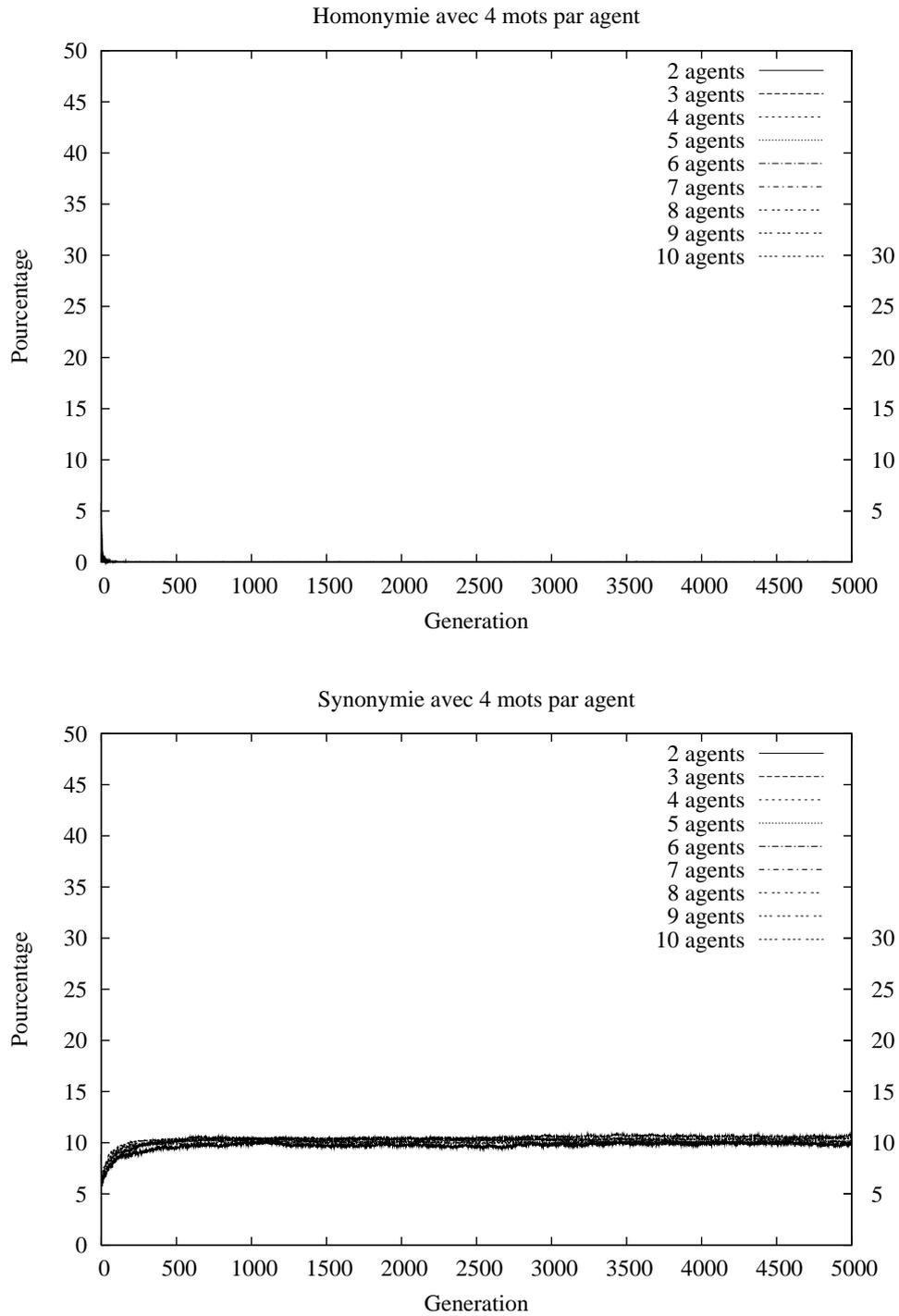


FIG. 6.12 – Homonymie et synonymie avec 4 mots et 4 sens possibles.

Notons au passage que le fait d'avoir plusieurs lexiques est un formidable potentiel pour permettre au système de se réadapter en cas d'impossibilité d'utilisation de certaines combinaisons lexicales.

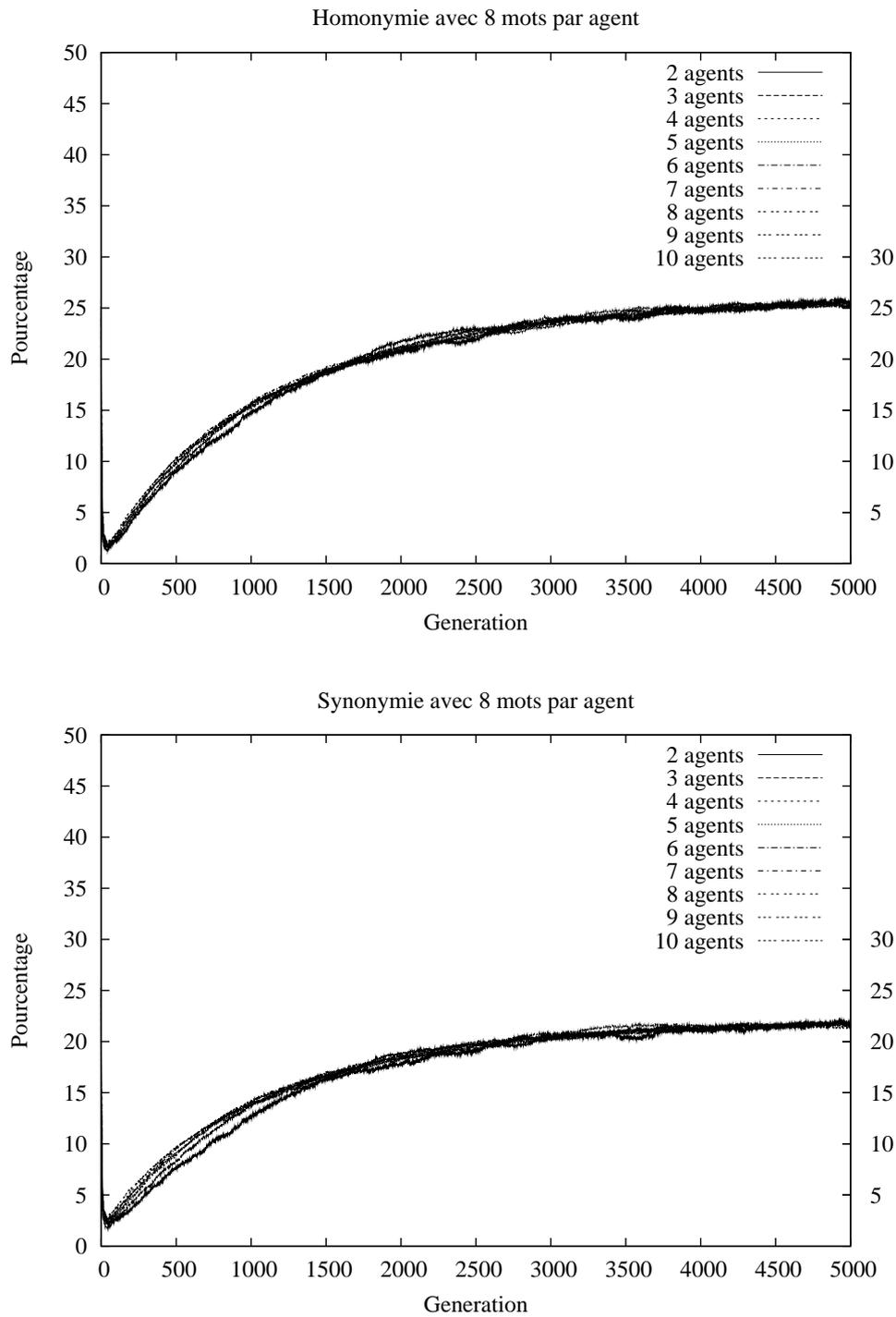


FIG. 6.13 – Homonymie et synonymie avec 8 mots et 8 sens possibles.

La figure 6.14 présente les résultats avec 16 mots.

Nous avons pu déjà observer que le taux maximum de communication est de 96,82% à la gé-

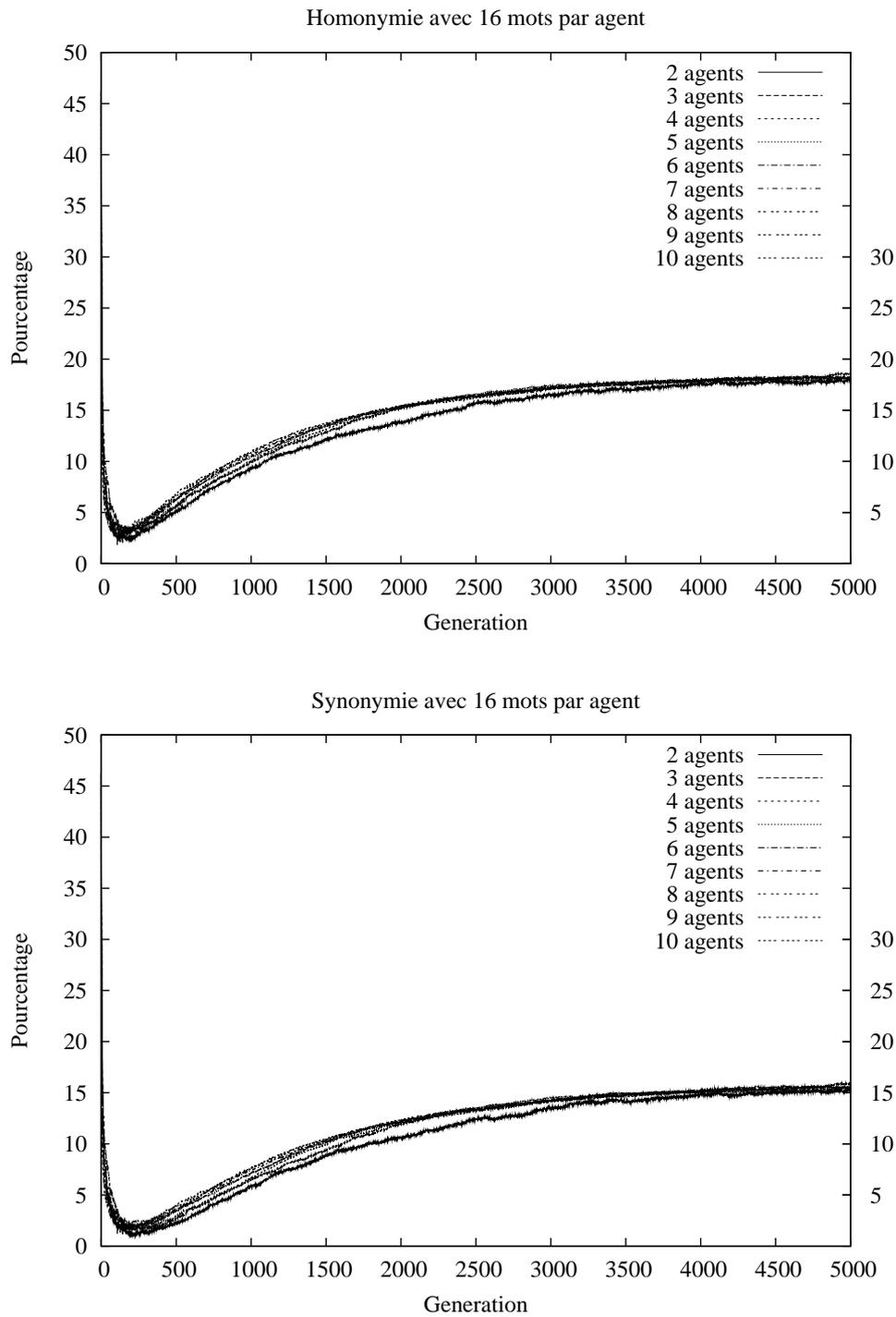


FIG. 6.14 – Homonymie et synonymie avec 16 mots et 16 sens possibles.

nération 108 tandis que le taux d'homonymie est alors de 1,86% et celui de synonymie de 1,21%. A cet instant, la valeur moyenne des agents est 28,47%. L'homonymie va alors progresser jusqu'à

un taux d'environ 18% et la synonymie jusqu'à 15,30% pendant que le taux de formation lexicale redescend aux environs de 75,70%. La valeur moyenne finale des agents est de 92,28%. Nous avons donc encore des confusions résiduelles. Les taux observés indiquent que les lexiques formés par les agents contiennent des confusions comme pour l'exemple à 8 mots. Cette confusion est moins importante mais reste latente. Si l'on suppose que les agents tentent encore de maintenir deux lexiques pour discuter avec un autre agent, il leur faut au moins 32 règles pour exprimer ces deux lexiques. Il devient donc alors impossible d'exprimer l'ensemble des sens avec seulement 12 règles par individu. Toutefois la confusion semble diminuer et une autre dynamique semble apparaître. Si l'on regarde de plus près, une différenciation commence à apparaître en fonction du nombre d'agents qui communiquent ensemble. On observe que la courbe à 2 agents semble un peu moins confuse que celle avec 3 agents et ainsi de suite, même si ce phénomène ne s'observe que vers la génération 2000. La tendance naturelle des systèmes de classeurs de type Pittsburgh simplifié à prévoir plusieurs choix non-déterministes semble être partiellement responsable de la confusion que nous venons d'observer. Le passage à 32 mots par agents devraient permettre d'empêcher le système d'être non-déterministe car le nombre de règles par individu devient alors insuffisant.

La figure 6.15 montre les résultats de l'évolution de l'homonymie et de la synonymie avec un lexique de taille 32.

Le seul moyen d'expliquer jusqu'à présent que les courbes des agents soient quasiment identiques était de supposer que chaque agent avait la possibilité de former un lexique, unique ou non avec chacun des autres agents avec lesquels il communique. Les résultats avec 32 mots montrent une forte différenciation dans les résultats. Comme précédemment, le taux d'homonymie est globalement plus élevé que celui de synonymie mais cette fois-ci le phénomène non-déterministe semble réduit à sa plus simple expression puisque le taux maximum d'homonymie est atteint avec deux agents et est de 7,96% tandis que le taux de synonymie est de 6,16%. Nous constatons que ces taux baissent lorsque le nombre d'agents augmente. Pourtant augmenter le nombre d'agents devraient rendre plus difficile la communication. C'est pour cette raison que les agents sont obligés de limiter au maximum leurs choix tout en arrivant à communiquer. Ainsi le non-déterminisme encore possible avec 2 agents, ne l'est plus avec 10 agents. On comprend alors mieux pourquoi la valeur moyenne des agents est de 92,42% pour la communication à 10 agents et seulement de 81,79% pour la communication à 2 agents.

Ces résultats confirment un phénomène que l'on observe dans la nature. Le gaspillage de ressources entraîne une moins bonne performance : la communication n'échappe pas à cette loi. Lorsque les fourmis communiquent de manière hétérogène, elles le font toujours de manière minimale, c'est à dire qu'elles tentent d'utiliser le moins de ressources possibles pour communiquer. C'est pour cette raison qu'elles utilisent un vocabulaire réduit pour que la communication avec les autres catégories de la communauté ait un coût minimal. Nos expérimentations ont confirmé ces observations. Forts de ce constat, nous avons souhaité savoir quelle pouvait être l'influence de l'observation d'un autre phénomène sur notre modèle de communication.

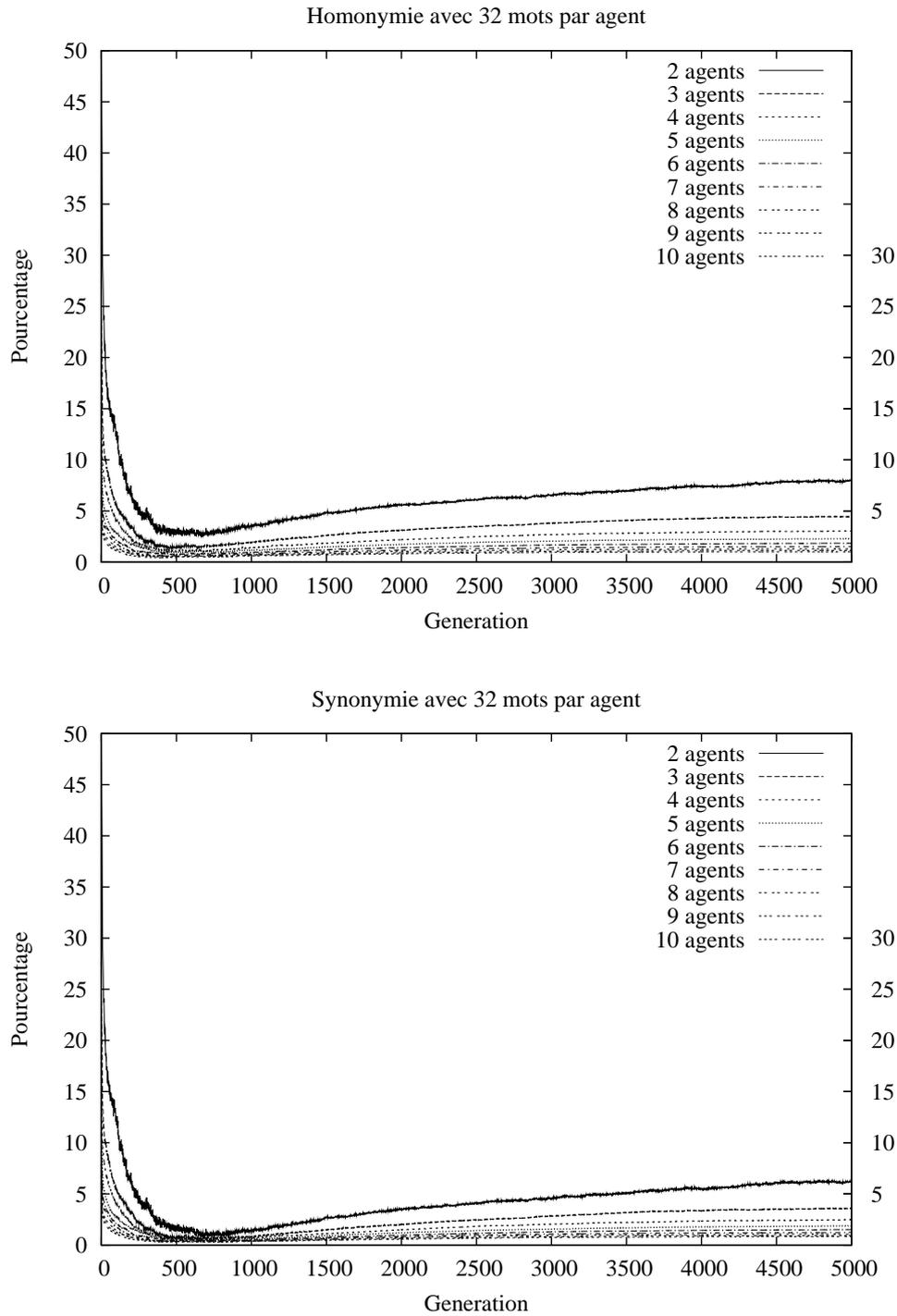


FIG. 6.15 – Homonymie et synonymie avec 32 mots et 32 sens possibles.

6.4.3 Combinaisons de lexiques

Nous avons fait de nouvelles expérimentations en tentant de faire communiquer deux communautés d'agents qui ont évolué séparément [Garfield, 1985]. Ce phénomène a été observé à de nom-

breuses reprises dans les ethnies qui ont peuplé l'Indonésie puis le Pacifique. On observe chez elles de nombreuses racines communes mais une incapacité à se comprendre. Nous profitons du cadre de l'éthologie synthétique [MacLennan et Burghardt, 1994] pour étudier trois scénarios de rencontres de deux communautés d'agents. Nous avons laissé les populations évoluer séparément pendant un certain nombre de générations puis nous avons choisi trois scénarios de rencontres qui correspondent à différents degrés d'hétérogénéité [Balch, 1997]. Le premier choix est le *diplomate*. Le diplomate est l'un des agents d'une communauté qui va discuter régulièrement avec son homologue dans l'autre communauté pour tenter de le comprendre. On peut imaginer que le diplomate est l'équivalent d'un explorateur qui désire apprendre la langue d'une tribu inconnue : le célèbre "problème Gavagai" de Willard Van Orman Quine (1960). Le second scénario est l'*ambassadeur*. L'ambassadeur est l'un des agents de la communauté qui s'installe dans l'autre communauté et qui communique à la fois avec sa communauté et l'autre communauté. L'ambassadeur communique davantage que le diplomate. Enfin le dernier scénario est la *fusion*. La fusion contraint tous les agents à communiquer les uns avec les autres. Nous allons étudier ces différents scénarios qui semblent une évolution naturelle de la rencontre de deux communautés.

Étude des 3 scénarios Nous avons choisi les paramètres qui se trouvent dans la table 6.16 pour l'expérimentation et dans la table 6.18 pour l'algorithme génétique. Nous avons moyenné nos expériences sur 30 expérimentations avec différentes racines aléatoires. Les agents sont répartis en deux groupes de 2, 3, 4 et 5 agents.

Paramètre	Réglage
Nombre d'agents	4, 6, 8 et 10
Début de la rencontre des communautés à la génération	500
Nombre de bits par mot et env. local	2 à 5
Nombre d'individus (<i>NombreIndividus</i>)	10
Nombre de règles par individu (<i>NombreRegles</i>)	12
Nombre d'essais entre générations	20

TAB. 6.16 – Paramètres du système multi-agents.

Nous présentons tout d'abord les résultats obtenus en terme de succès de communication avec les trois méthodes et un lexique de taille 4 : ce sont les figures 6.16, 6.17 et 6.18 qui présentent ces résultats.

Paramètre	Réglage
$P_{\#}$	0%
$P_{\text{Croisement}}$	70%
P_{Mutation}	0, 1%
Elitisme	$\frac{2}{10} \times 100 = 20\%$
Mécanisme de Sélection	Roulette Wheel
Sélection des règles (cf. § 4.3.3)	Spécifique
Croisement	Monopoint
Nombre de Générations	2000

TAB. 6.18 – Paramètres de l’algorithme génétique.

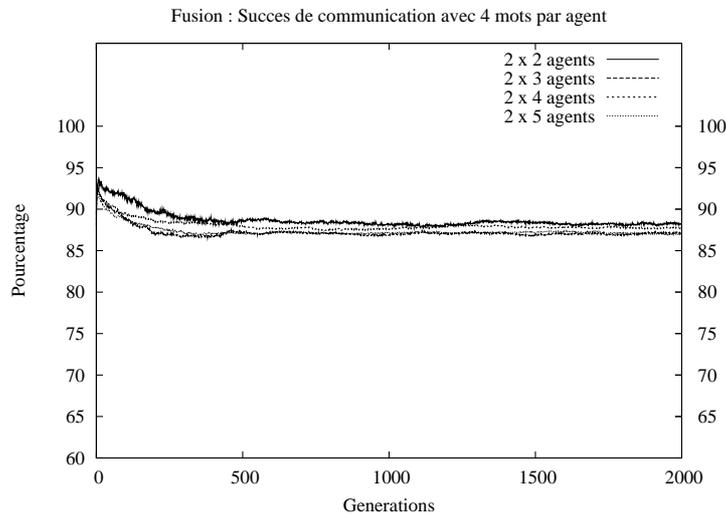


FIG. 6.16 – Succès de communication pour le scénario “Fusion” avec 4 mots.

La fusion des deux communautés ne perturbe en rien l’évolution du lexique à ce stade. Nous obtenons des résultats proches de ceux obtenus précédemment avec 4 mots et 4 environnements locaux possibles. Par contre, l’introduction d’un ambassadeur dans chacune des communautés perturbe fortement l’évolution du lexique qui se met alors à osciller autour de la courbe obtenue pour la fusion, mais qui reste en moyenne proche de celle-ci. Quant au diplomate, comme l’ambassadeur, il perturbe fortement l’évolution du lexique mais cette fois-ci de manière légèrement positive puisque l’expérience avec le diplomate termine 0,43% au dessus de celle avec l’ambassadeur. La figure 6.19 explique ce résultat.

Le taux d’homonymie diffère entre l’ambassadeur et le diplomate. Pour ce dernier, le taux d’homonymie oscille légèrement au dessus des 0% tandis que celui de l’ambassadeur est d’environ 3%. L’étude de la synonymie ne révèle pas des écarts aussi significatifs, celle-ci étant la même que celle

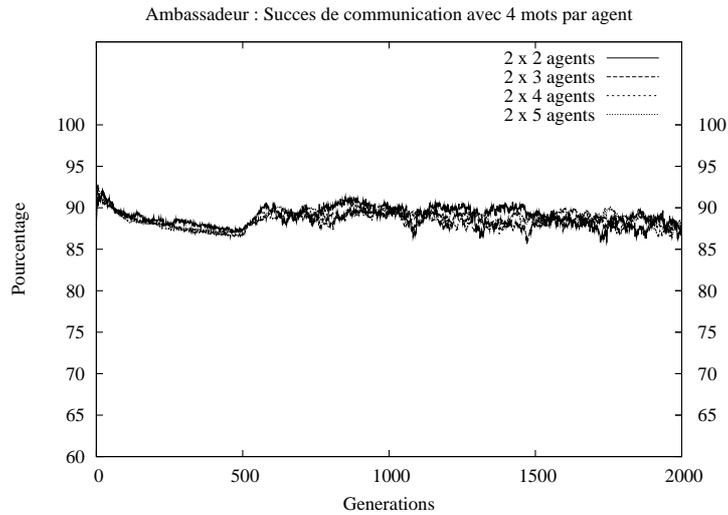


FIG. 6.17 – Succès de communication pour le scénario “Ambassadeur” avec 4 mots.

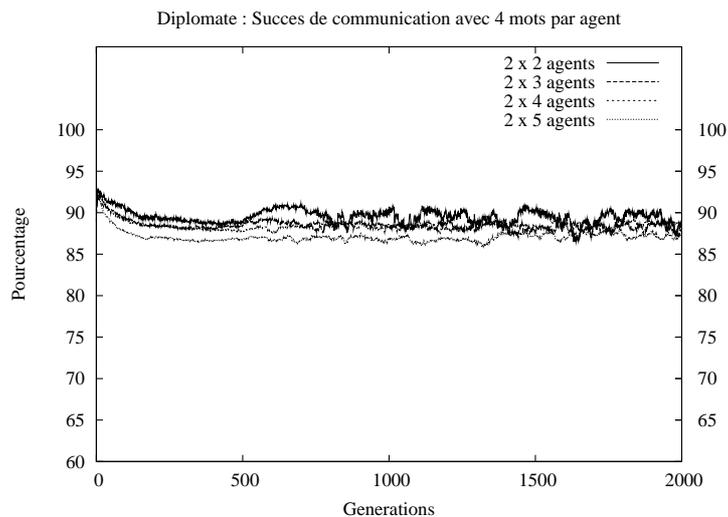


FIG. 6.18 – Succès de communication pour le scénario “Diplomate” avec 4 mots.

observée dans les expérimentations de la section 6.4.2. Cet écart de près de 3% dans le taux d’homonymie justifie un écart de 0,43% dans le succès de communication. Pour les petits lexiques et avec trop de ressources, le diplomate est légèrement plus efficace pour gérer l’absorption du langage. Voyons maintenant le cas avec un lexique de taille 8 que nous savons plus difficile.

Nous pouvons trouver les résultats pour 8 mots et 8 environnements locaux sur les figures 6.20, 6.21 et 6.22.

Comme précédemment, la fusion des deux communautés n’entraîne aucune variation dans la formation du lexique. Les résultats sont comparables à ceux obtenus avec une seule communauté et 8

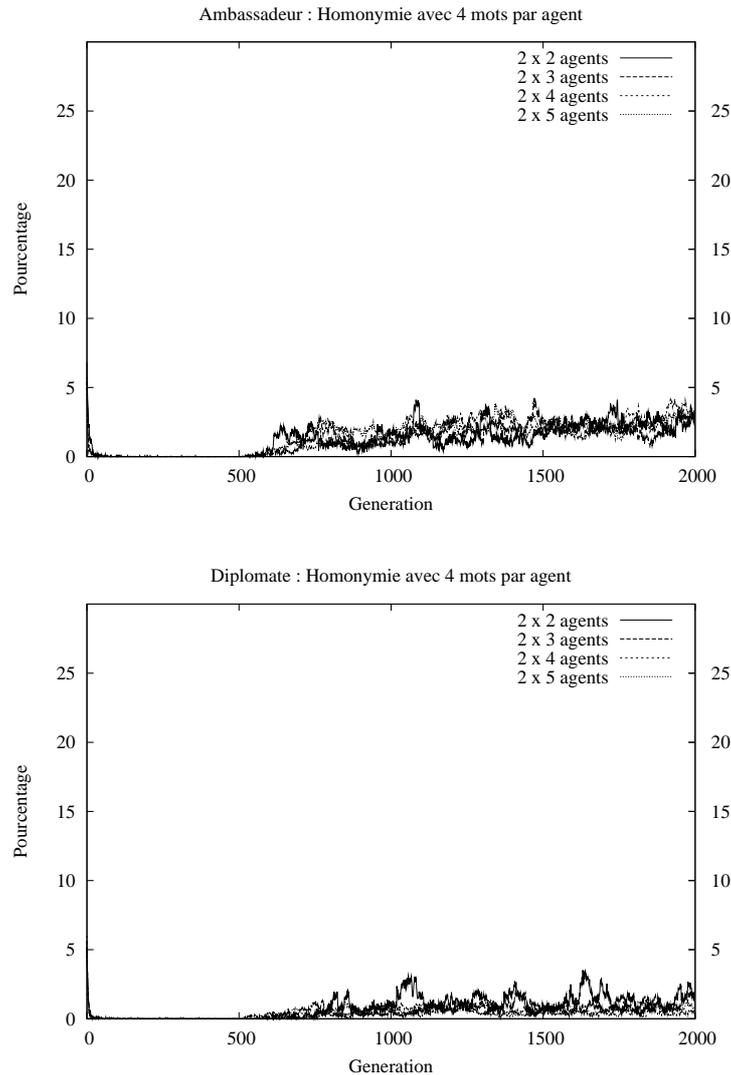


FIG. 6.19 – Homonymie chez l’ambassadeur et le diplomate utilisant 4 mots.

mots possibles. Cette fois-ci, c’est l’ambassadeur qui prend l’avantage sur le diplomate. Comme nous l’avons vu auparavant, nous atteignons avec 8 mots, un seuil qui précède une utilisation plus grande des ressources disponibles ainsi qu’une différenciation des résultats en fonction du nombre d’agents. Ce seuil permet à l’ambassadeur d’améliorer la communication d’environ 4,78% par rapport à la fusion. Le diplomate améliore également la communication mais de seulement 2,81% en moyenne. Dans les deux cas les gains sont intéressants et s’expliquent d’une part par les figures 6.23, 6.24 et 6.25 et d’autre part par les figures 6.26, 6.27 et 6.28.

En effet, lorsqu’on regarde les pourcentages de synonymie, on constate que l’ambassadeur diminue de 3,75% la synonymie par rapport à la fusion et de 2,08% par rapport au diplomate. La synonymie n’explique pas à elle seule le gain en succès de communication. En regardant l’évolution

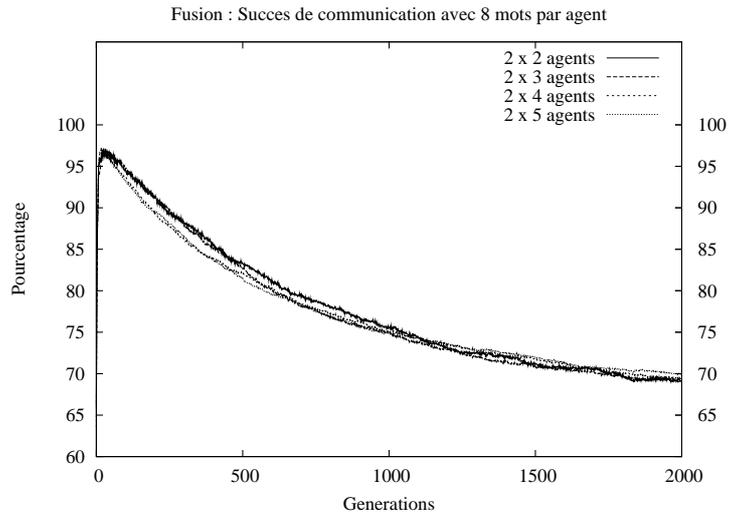


FIG. 6.20 – Succès de communication pour le scénario “Fusion” avec 8 mots.

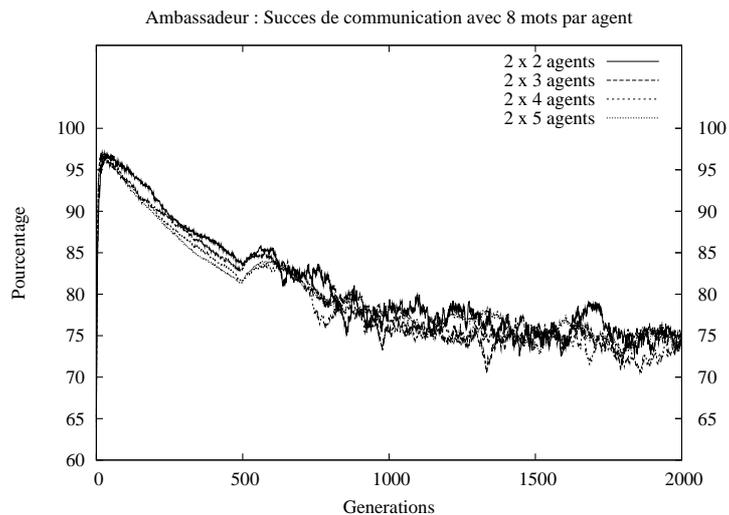


FIG. 6.21 – Succès de communication pour le scénario “Ambassadeur” avec 8 mots.

de l’homonymie, nous pouvons voir qu’avec l’ambassadeur, la diminution de l’homonymie par rapport à la fusion est d’environ 3,35%. Avec le diplomate cette diminution est de 1,98%. La méthode de l’ambassadeur est donc plus intéressante avec 8 mots et permet de maintenir un seuil de communication acceptable. Il semble donc que nos deux méthodes utilisent le surplus de ressources disponibles, ce qui entraîne une amélioration des résultats observés. Toutefois le diplomate semble consommer un peu moins de ressources que la méthode ambassadeur. Regardons maintenant ce qui se passe lorsque l’on passe à un lexique de taille 16.

Les figures 6.29, 6.30 et 6.31 contiennent les graphiques des 3 scénarios pour 16 mots et 16

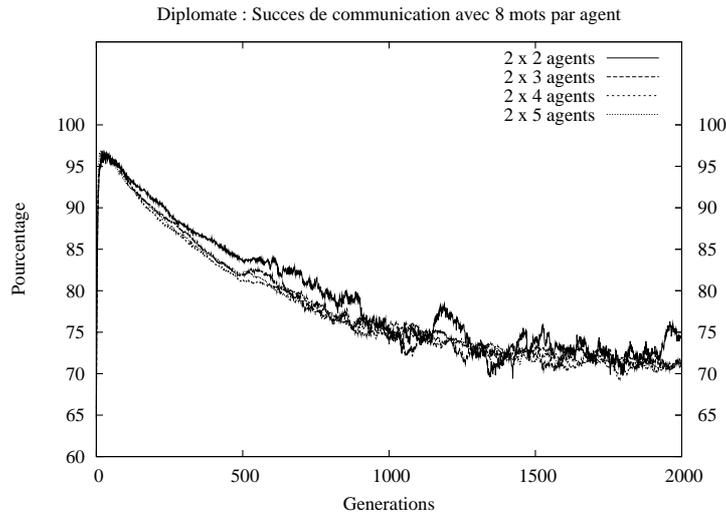


FIG. 6.22 – Succès de communication pour le scénario “Diplomate” avec 8 mots.

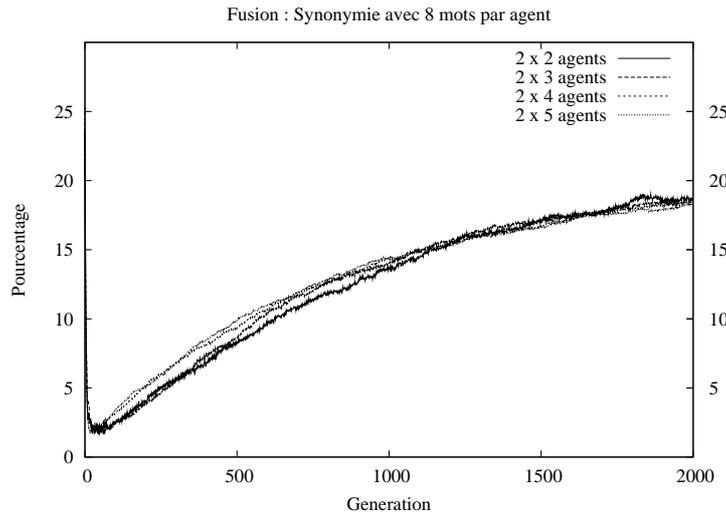


FIG. 6.23 – Synonymie dans le scénario “Fusion” en utilisant 8 mots.

environnements locaux.

Nous pouvons constater une fois encore que la fusion se passe sans aucun problème et que cette courbe suit les résultats obtenus avec une seule communauté. Pour l’ambassadeur, il en est autrement puisque nous constatons une perte de 3,78% du succès de communication. Le diplomate ne cède quant à lui que 2,13%. Si l’on regarde les pertes en termes de synonymie, le diplomate perd 1,47% et l’ambassadeur 2,13%. En terme d’homonymie, ces pertes sont de 1,10% pour le diplomate et de 1,50% pour l’ambassadeur. Si l’on observe attentivement les courbes des figures 6.32 et 6.33, nous pouvons voir que les courbes s’adoucissent alors que le nombre d’agents augmente dans chaque

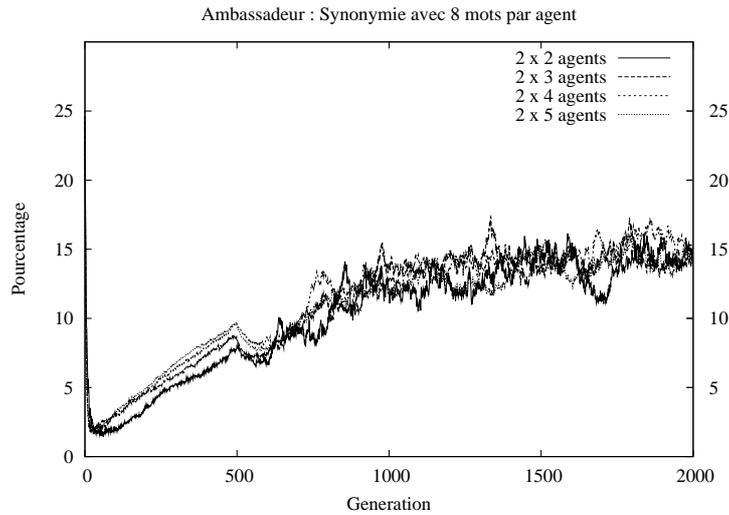


FIG. 6.24 – Synonymie dans le scénario “Ambassadeur” en utilisant 8 mots.

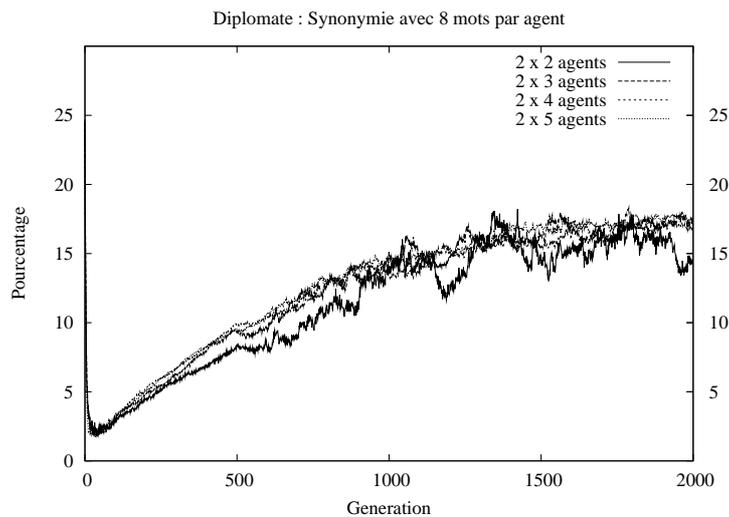


FIG. 6.25 – Synonymie dans le scénario “Diplomate” en utilisant 8 mots.

communauté. Nous voyons également que ce phénomène est plus important avec le diplomate. Ce résultat semble cohérent puisque le diplomate interagit dans l’autre communauté seulement avec un autre diplomate, par conséquent les interactions qu’il peut avoir ont une moindre influence lorsque le diplomate représente une part moindre de la population. En effet, dans les communautés de 2 agents, le diplomate ou l’ambassadeur représente 50% de la communauté, sa façon de communiquer est donc très importante. Lorsque la communauté est de 5 agents, le diplomate ne représente alors plus que 20% de la communauté. L’ambassadeur quant à lui interagit avec sa communauté et l’ensemble de l’autre communauté. Si la communication est difficile, cela se ressentira donc plus fortement qu’avec le

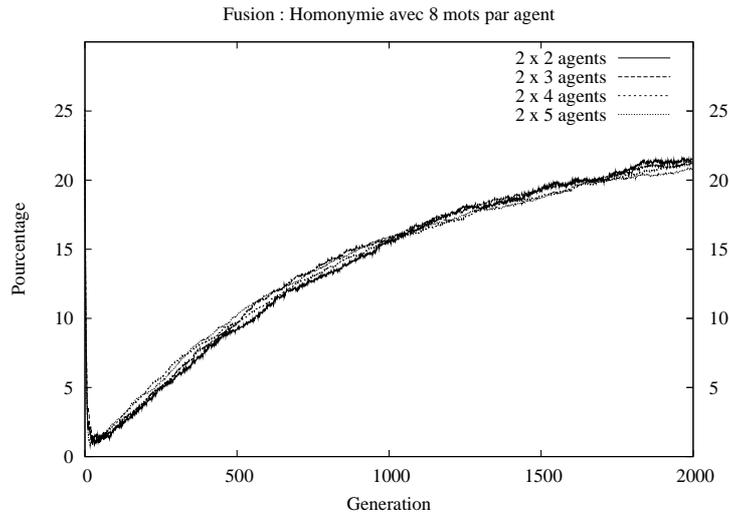


FIG. 6.26 – Homonymie dans le scénario “Fusion” en utilisant 8 mots.

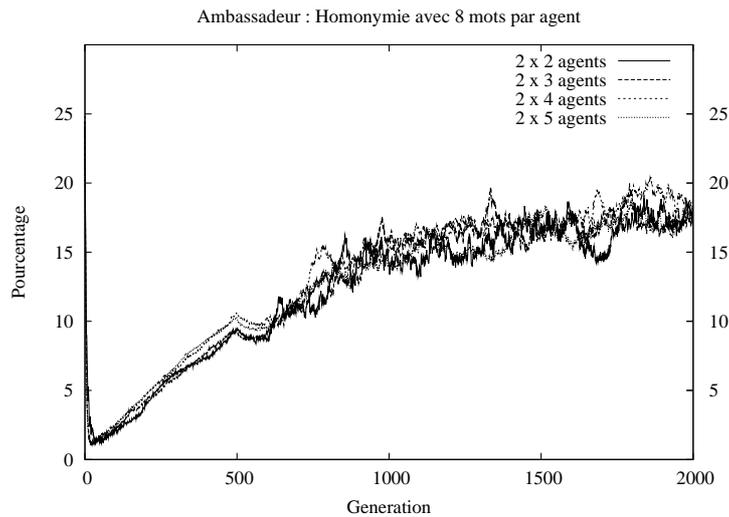


FIG. 6.27 – Homonymie dans le scénario “Ambassadeur” en utilisant 8 mots.

diplomate mais moins fortement si la communauté est grande. Regardons l'évolution de ce phénomène avec le passage à un lexique de taille 32.

Les figures 6.34, 6.35 et 6.36 présentent les résultats obtenus pour les trois scénarios avec 32 mots et 32 environnements locaux possibles.

Les résultats obtenus avec la fusion sont plus de 2% en dessous de la plus mauvaise courbe à 32 mots des expériences à une communauté. Contrairement à ce que nous avons observé pour cette même expérience à une seule communauté, les courbes ne se distinguent pas en fonction du nombre d'agents. La fusion semble avoir un coût, même si celui-ci est très faible, puisqu'on n'observe un

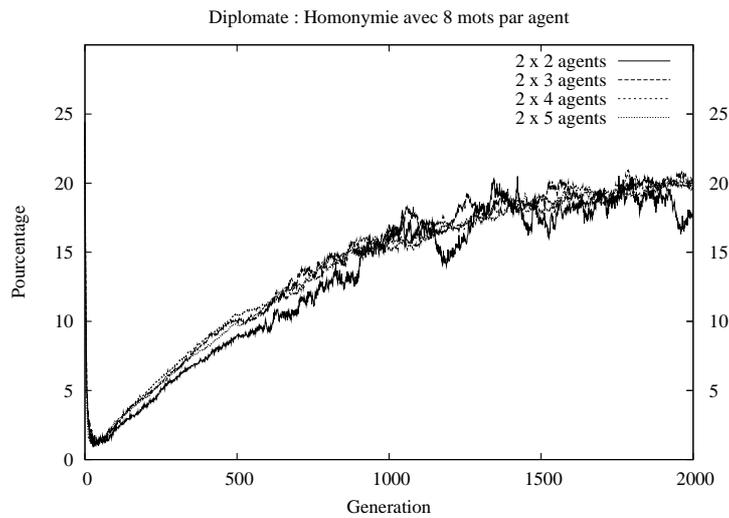


FIG. 6.28 – Homonymie dans le scénario “Diplomate” en utilisant 8 mots.

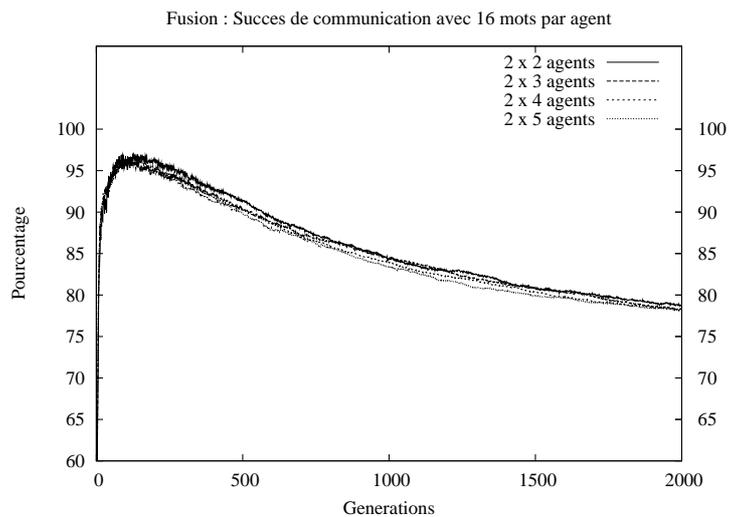


FIG. 6.29 – Succès de communication pour les trois scénarios avec 16 mots.

changement qu’avec 32 mots et 32 environnements locaux possibles. Si l’on regarde le comportement du scénario ambassadeur, nous observons un véritable effondrement du succès de la communication. Le coût en ressource qu’engendre l’ambassadeur se fait logiquement et fortement ressentir lorsqu’on passe à un lexique de taille 32 car l’ensemble des ressources est déjà monopolisé afin de permettre la formation d’un lexique dans chacune des communautés. Lorsque ces communautés se rencontrent, leur lexique est en général extrêmement différent ce qui explique que le succès de communication s’effondre. Aucune différence n’est visible au niveau de la taille de chaque communauté. L’effondrement constaté approche les 20%. Pour le diplomate, comme nous l’avons déjà observé, le coût de

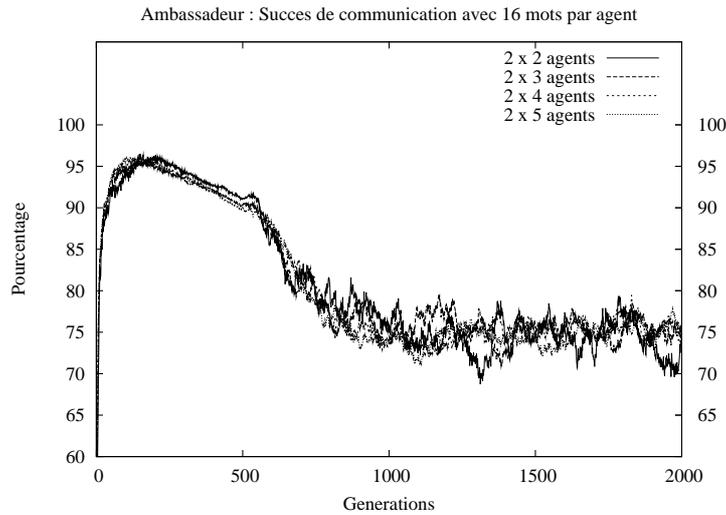


FIG. 6.30 – Succès de communication pour le scénario “Ambassadeur” avec 16 mots.

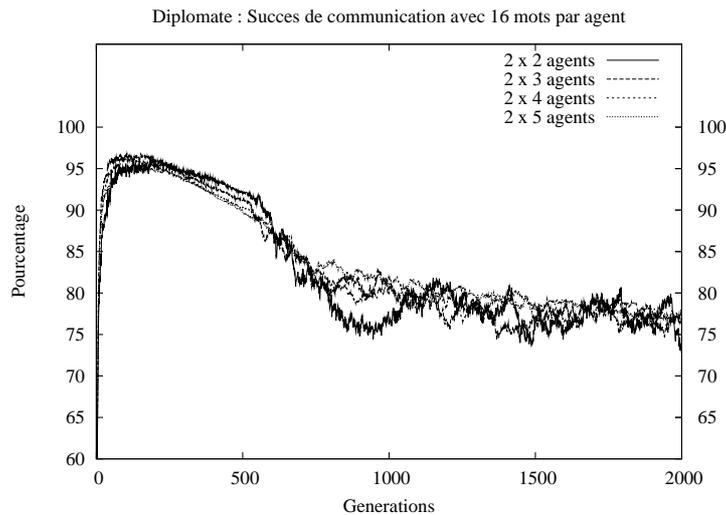


FIG. 6.31 – Succès de communication pour le scénario “Diplomate” avec 16 mots.

cette communication est plus faible que celui de l’ambassadeur. Malgré cela, et comme pour l’ambassadeur, les diplomates arrivent au moment le moins opportun puisque les ressources sont totalement attribuées à la formation d’un lexique dans chacune des communautés. Il en résulte une baisse de succès de communication de 8,66%. Par contre le diplomate est sous l’influence du nombre d’agents. Nous avons déjà constaté ce phénomène lorsque nous n’avions qu’une seule communauté. Le scénario du diplomate est donc le seul qui permette d’obtenir un comportement proche de celui que nous avons observé. Les figures 6.37, 6.38 et 6.39 et les figures 6.40, 6.41 et 6.42 permettent de vérifier que le diplomate est la seule méthode qui permette aux agents de se mélanger tout en gardant leur

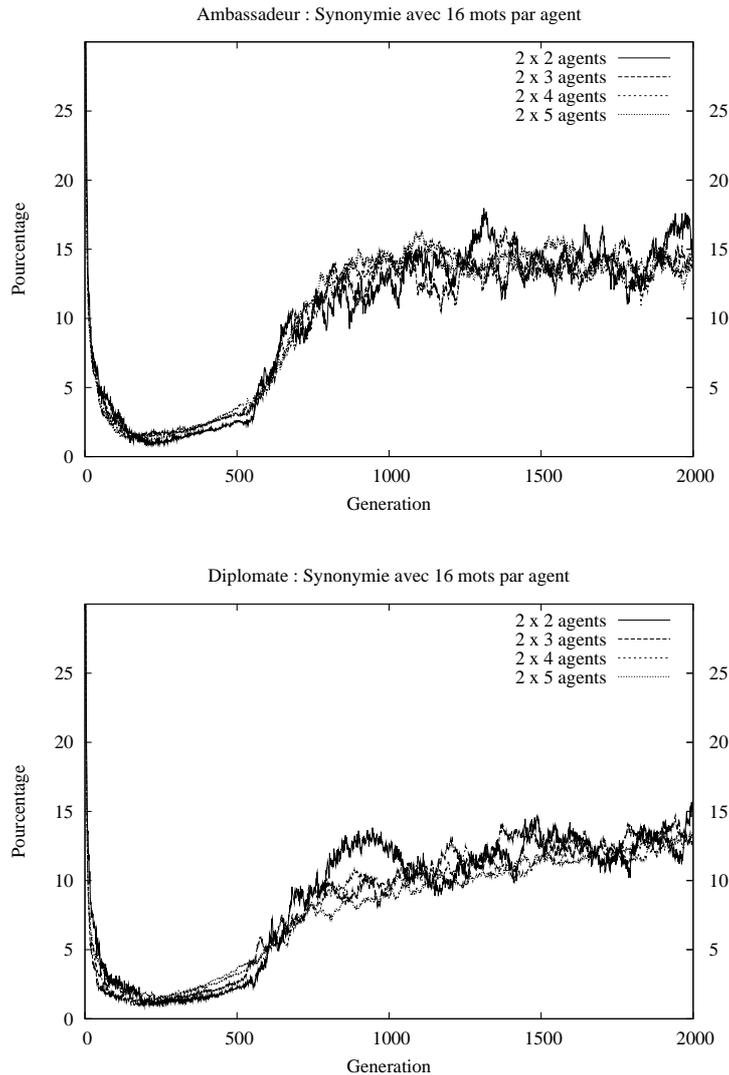


FIG. 6.32 – Synonymie pour l’ambassadeur et le diplomate avec 16 mots.

identité communautaire. Nous observons en effet que c’est le seul scénario avec lequel synonymie et homonymie se comportent comme s’il n’y avait qu’une seule communauté. Elle n’est pas pour autant la meilleure méthode pour permettre aux langues de se mélanger pour créer un nouveau lexique compréhensible des deux communautés communicantes.

Nous l’aurons compris, la fusion est certainement la méthode la moins coûteuse pour permettre à deux communautés de se rencontrer. Toutefois cette méthode a des limites puisque lorsque la communication utilise de manière optimale les ressources disponibles, la fusion entraîne une diminution du succès de communication et fait perdre toute autonomie aux communautés puisque celles-ci ne forment plus qu’une seule communauté. Le fait que la fusion se produise relativement tôt dans l’évolution est certainement un facteur déterminant dans sa stabilité.

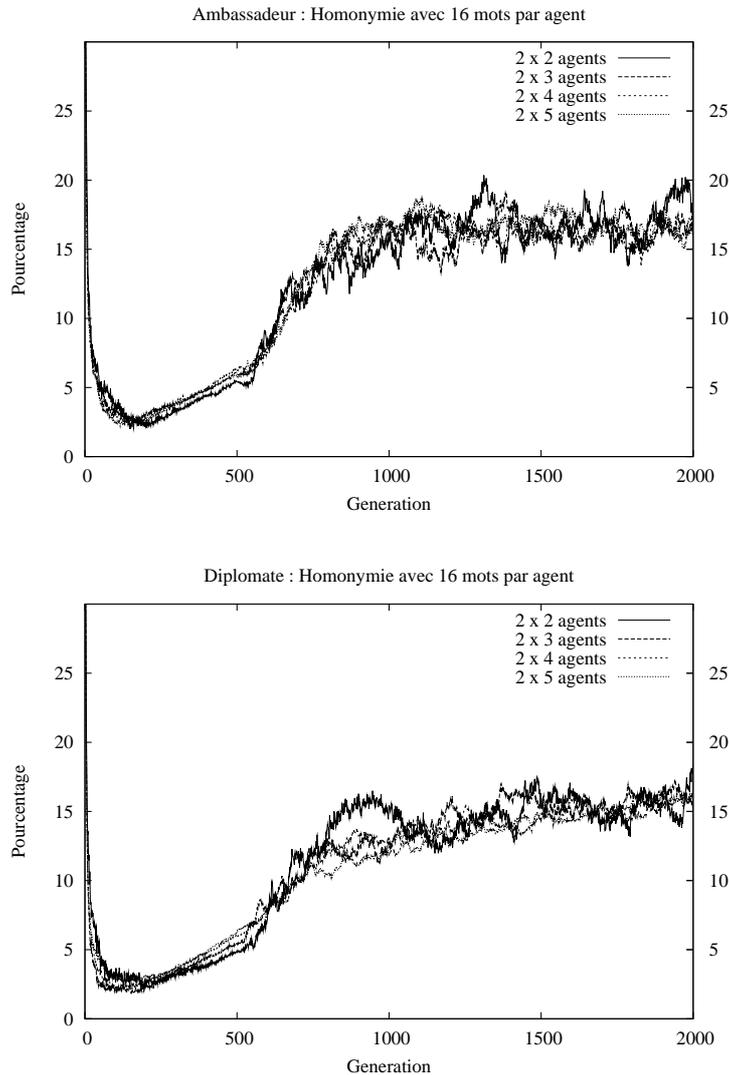


FIG. 6.33 – Homonymie pour l’ambassadeur et le diplomate avec 16 mots.

La méthode du diplomate est avantageuse dans la mesure où la communauté à laquelle il appartient est suffisamment grande et où l’on utilise de manière optimale les ressources disponibles. Ce n’est encore pas la meilleure méthode pour rapprocher deux communautés mais elle permet à chacune d’entre elles de conserver le plus longtemps possible leur identité.

Enfin la méthode de l’ambassadeur est un intermédiaire entre la fusion et le diplomate qui nécessite que l’on ait un surplus de ressources pour fonctionner correctement.

Cette étude nous a donné l’idée de créer un dernier scénario dans lequel les deux communautés vont utiliser ses différents scénarios tour à tour.

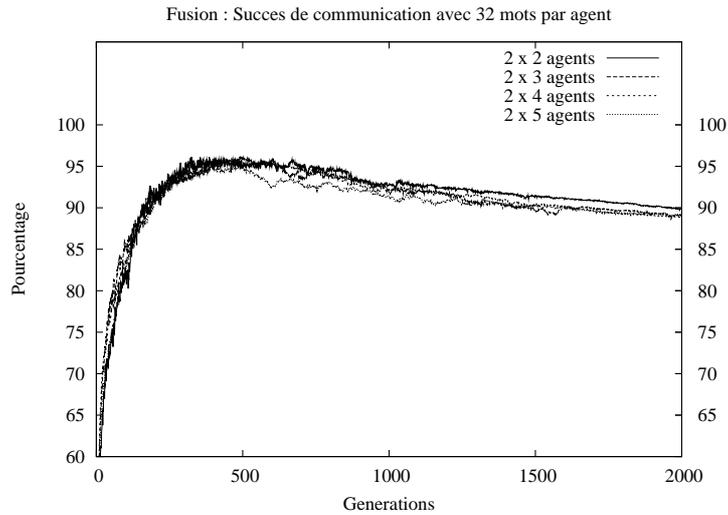


FIG. 6.34 – Succès de communication pour les trois scénarios avec 32 mots.

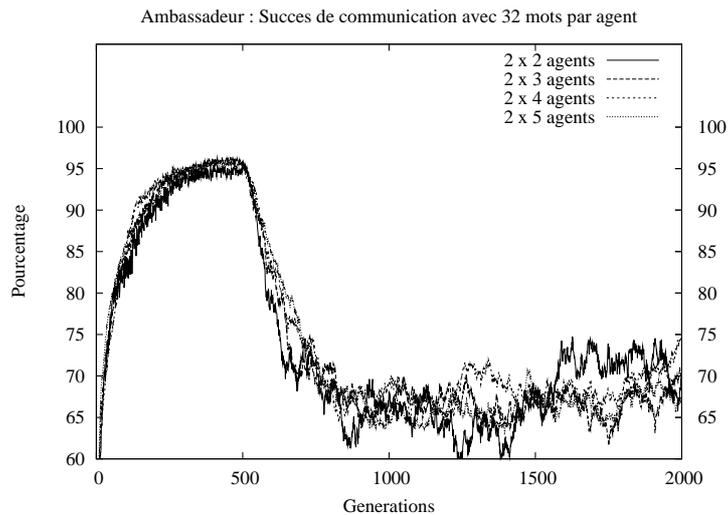


FIG. 6.35 – Succès de communication pour le scénario “Ambassadeur” avec 32 mots.

La rencontre des 2 mondes Nous allons vérifier nos conclusions à l’aide d’une dernière série d’expérimentations pour lesquelles nous avons gardé les mêmes paramètres. La seule différence se situe au niveau de la façon d’utiliser les différents scénarios. Toutes les 500 générations nous changerons de scénarios en commençant par une phase séparée comme pour les expériences précédentes. Ensuite, nous avons imaginé que deux diplomates se rencontraient comme l’explorateur du problème Gavagai rencontre l’indigène. Nos scénarios ne prennent pas en compte la malencontreuse disparition d’un diplomate. Puis une fois le contact établi, un ambassadeur se rend dans le pays de son homologue pour augmenter les échanges et enfin les deux communautés finissent par fusionner durant les 500

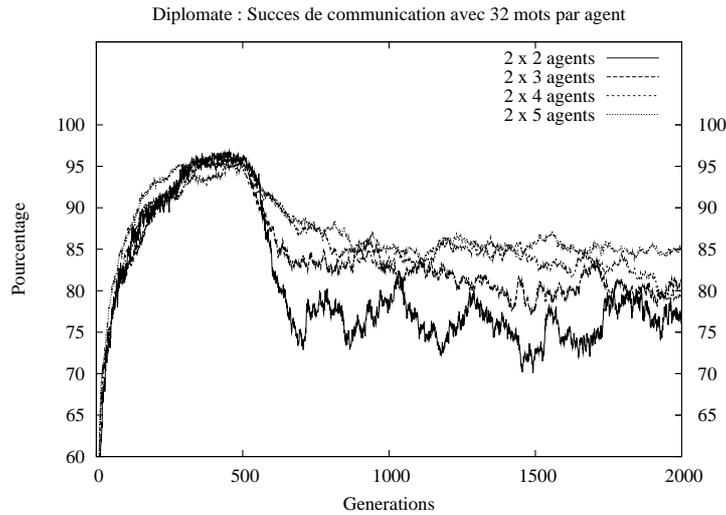


FIG. 6.36 – Succès de communication pour le scénario “Diplomate” avec 32 mots.

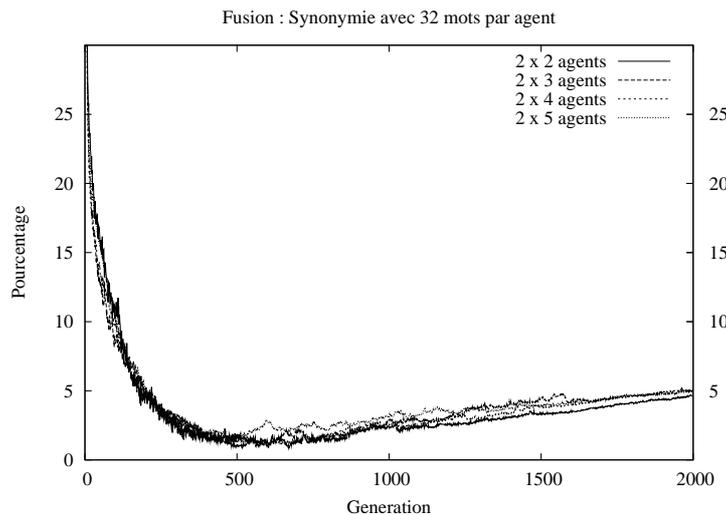


FIG. 6.37 – Synonymie dans les trois scénarios en utilisant 32 mots.

dernières générations. La figure 6.43 montre les résultats obtenus avec un lexique de taille 4.

Nous observons qu’après une phase sans chute d’oscillations dues au diplomate puis à l’ambassadeur, le succès de communication se stabilise à 86,83%, soit des résultats similaires à ceux observés dans la figure 6.8. Notre solution n’apporte donc rien par rapport à l’une des solutions prise séparément. Il faut regarder les résultats obtenus lors du passage à un lexique de taille 8.

Ces résultats se trouvent dans la figure 6.44.

Lorsqu’on regarde les résultats en détails on constate un gain de 0,55% en succès de communication avec notre solution par rapport à une solution de fusion classique. Le résultat reste en des-

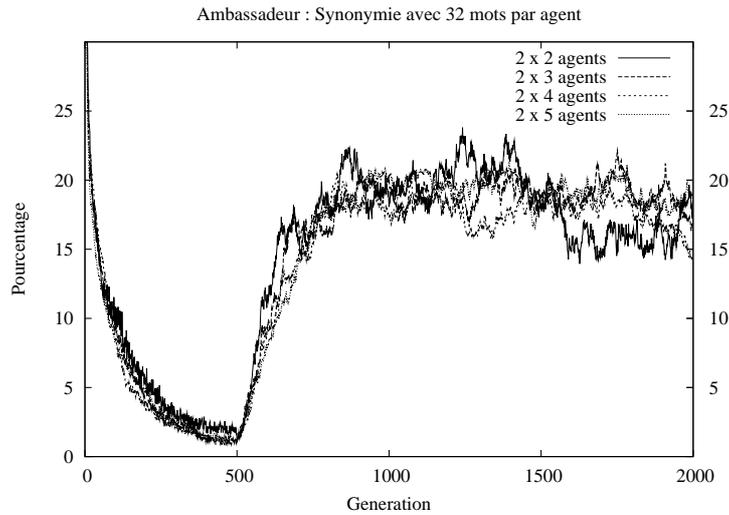


FIG. 6.38 – Synonymie dans le scénario “Ambassadeur” en utilisant 32 mots.

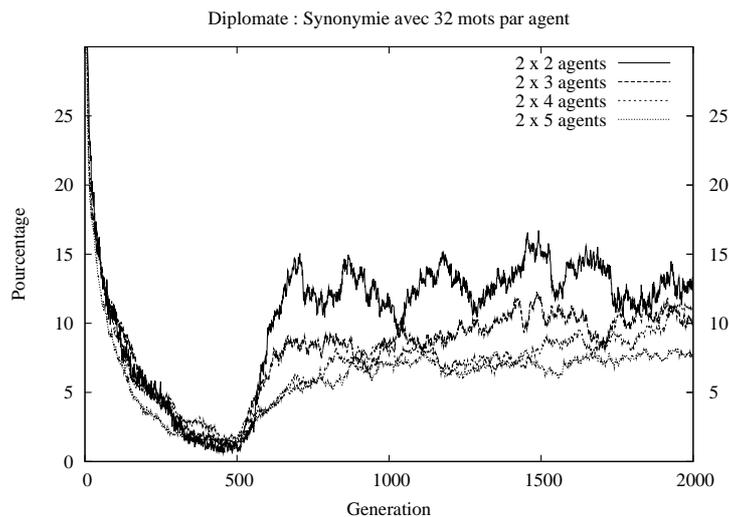


FIG. 6.39 – Synonymie dans le scénario “Diplomate” en utilisant 32 mots.

sous de ce que l’on peut obtenir avec le diplomate ou l’ambassadeur. Les résultats obtenus jusqu’ici confirment que notre approche n’a d’intérêt que lorsque les ressources nécessaires à la communication sont convenablement utilisées. Nous allons vérifier cela avec le passage à 16 mots et 16 environnements locaux.

La figure 6.45 contient les résultats pour un lexique de taille 16.

Cette fois-ci, notre approche est meilleure que les autres prises séparément (cf. figure 6.29). Au final, nous obtenons un succès de communication de 79,90% soit 1,57% de mieux que la fusion qui était le meilleur résultat. Notre hypothèse se vérifie pour le moment. Regardons si pour le passage à

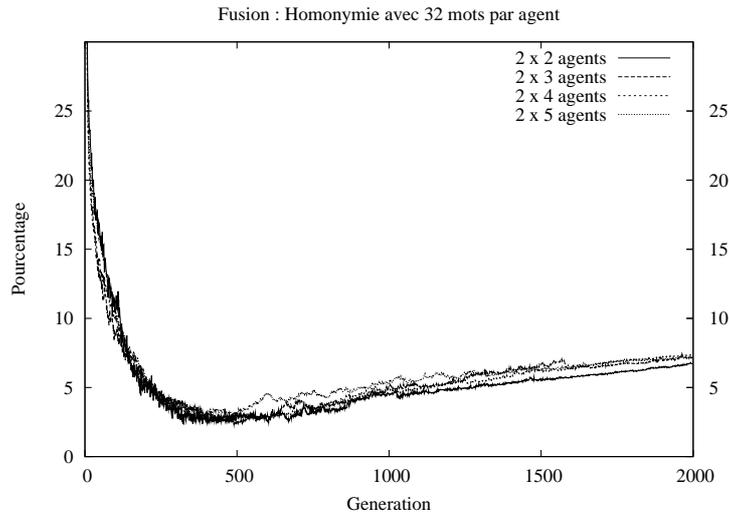


FIG. 6.40 – Homonymie dans les trois scénarios en utilisant 32 mots.

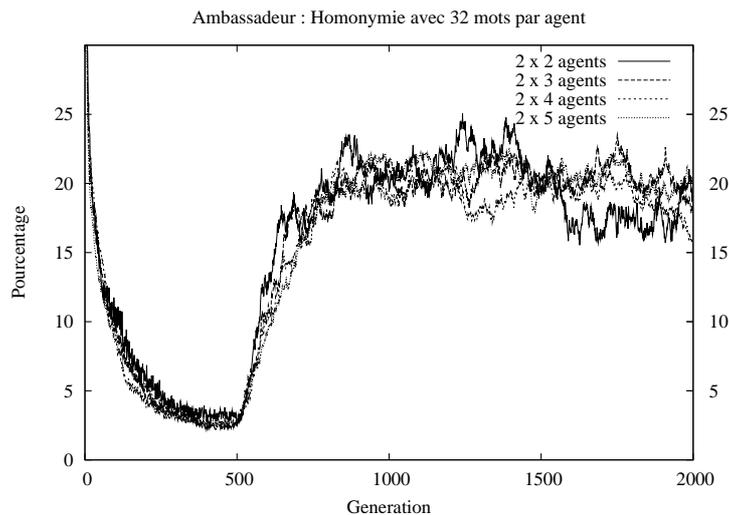


FIG. 6.41 – Homonymie dans le scénario “Ambassadeur” en utilisant 32 mots.

un lexique de taille 32 nous obtenons toujours de bons résultats.

La figure 6.46 présente ces résultats.

Nous obtenons à la fin de l'évolution un taux de formation lexical de 90,14% soit 0,92% de mieux que la meilleure des courbes pour les scénarios séparés. Cet effort de 0,92% est certainement plus difficile à faire que celui de 1,57% car nous dépassons 90% de réussite soit un lexique quasiment parfait. D'autre part avec 32 mots, il y a moins de ressources à gaspiller qu'avec 16, l'ambassadeur est donc légèrement moins efficace dans son rôle comme nous l'avions pressenti.

Notre dernière approche était basée sur un phénomène concret qui est que lorsque deux commu-

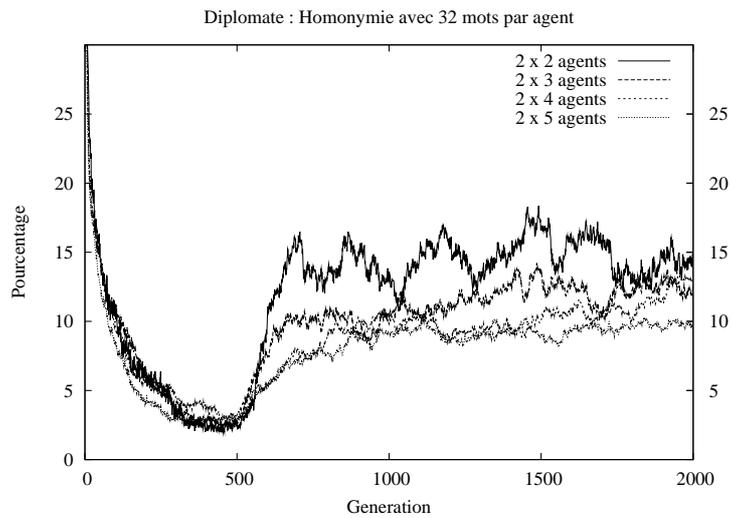


FIG. 6.42 – Homonymie dans le scénario “Diplomate” en utilisant 32 mots.

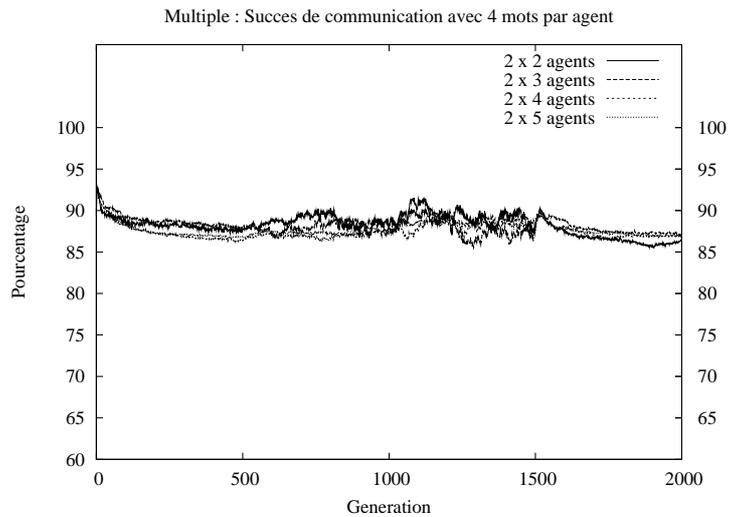


FIG. 6.43 – Succès de communication pour la rencontre des 2 mondes avec 4 mots.

nautés se rencontrent, ce sont d’abord deux individus, les diplomates, l’explorateur et le chasseur, qui apprennent à se comprendre. Une fois cette phase terminée, le diplomate peut s’installer avec les rudiments de langage qu’il a appris dans le pays de ses voisins. Enfin lorsque l’entente devient grande, les deux communautés peuvent fusionner. La formation d’un langage suit donc une logique dans la manière dont il est utilisé dans cette dernière approche [Dessalles, 1998a, Dessalles, 1998b].

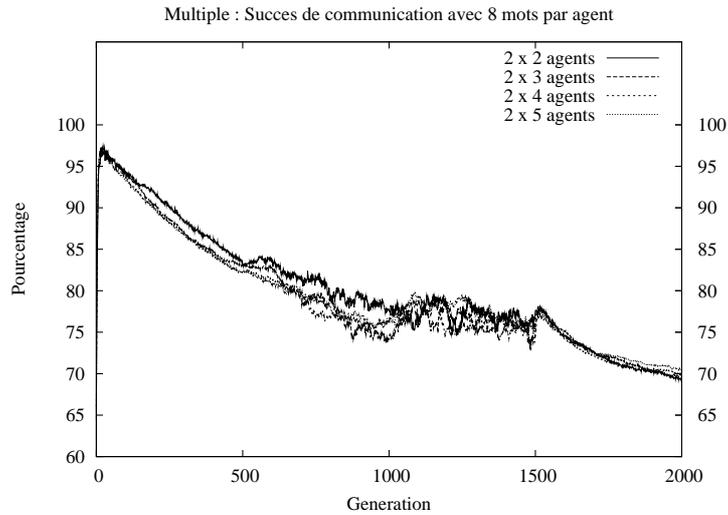


FIG. 6.44 – Succès de communication pour la rencontre des 2 mondes avec 8 mots.

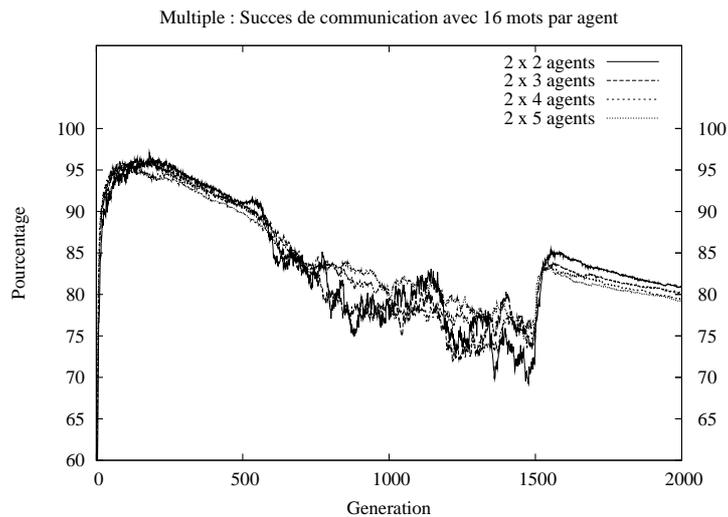


FIG. 6.45 – Succès de communication pour la rencontre des 2 mondes avec 16 mots.

6.5 Conclusion

Nous avons donc pu vérifier que nos hypothèses et notre outil étaient cohérents avec ce qui peut sembler naturel dans le cours de l'évolution de systèmes multi-agents hétérogènes. Le modèle minimal de communication a montré qu'il était utilisable avec un modèle ancré dans la réalité et avec des systèmes apprenants réactifs étant capables d'utiliser une chaîne binaire : c'est le cas des systèmes de classeurs Pittsburgh simplifié et XCS.

Nous avons pu observer qu'il était nécessaire d'utiliser pleinement les ressources de communication disponibles pour que la communication soit efficace entre agents hétérogènes. Ensuite, nous

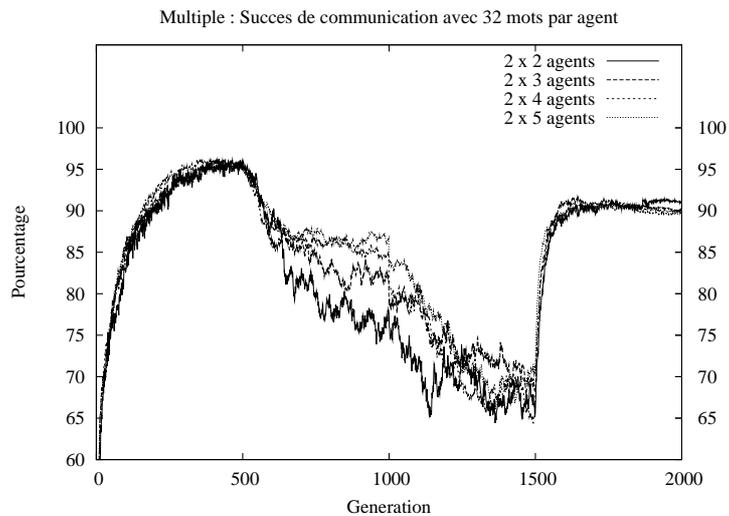


FIG. 6.46 – Succès de communication pour la rencontre des 2 mondes avec 32 mots.

avons vu que l’homonymie et la synonymie étaient des phénomènes naturels courants qui apparaissent avec le surplus de ressources puis s’atténuent avec l’augmentation du nombre d’agents qui communiquent. Nous avons également vu que plus le nombre d’agents est grand et plus l’usage des ressources est élevé, et plus la communication est efficace.

Enfin nous avons créé deux communautés d’agents hétérogènes qui vivent séparément dans un premier temps. Puis, nous avons imaginé trois scénarios. La fusion qui fait fusionner l’ensemble des deux communautés en faisant communiquer totalement les agents. Le diplomate est l’un des agents de chaque communauté qui va parlementer avec son homologue diplomate et enfin l’ambassadeur qui est la version implantée du diplomate. Par conséquent, l’ambassadeur interagit avec les deux communautés en permanence. Nous avons pu observer que la fusion est la méthode la moins coûteuse pour permettre à deux communautés de se rencontrer. Toutefois cette méthode a des limites puisque lorsque la communication utilise de manière optimale les ressources disponibles, la fusion entraîne une diminution du succès de communication et fait perdre leurs origines lexicales aux communautés qui fusionnent. Le fait qu’elle se produise relativement tôt dans l’évolution est un facteur déterminant dans sa stabilité.

La méthode du diplomate est avantageuse dans la mesure où la communauté à laquelle il appartient est suffisamment grande et où l’on utilise de manière optimale les ressources communicantes disponibles. Ce n’est encore pas la meilleure méthode pour rapprocher deux communautés mais elle permet à chacune d’entre elle de conserver le plus longtemps possible son identité.

Enfin la méthode de l’ambassadeur est un intermédiaire entre la fusion et le diplomate et préférera que l’on est trop de ressources pour bien fonctionner.

Nous avons vérifié ce constat en créant une dernière série d’expérimentations dans lesquelles le diplomate puis l’ambassadeur et enfin la fusion ont lieu après la première phase de vie séparée. Ces

expérimentations ont permis de confirmer l'intérêt de chacun des scénarios présentés et d'observer que la taille du lexique est un facteur déterminant dans la réussite de la rencontre de deux mondes.

Chapitre 7

Conclusion

7.1 Bilan

Nous avons étudié dans un premier temps (chapitre 2) les différents types de systèmes multi-agents (SMA) en nous inspirant de la nature à travers, entre autres, les fourmis, les étourneaux et les singes. Nous avons également observé pourquoi la nature utilise la communication ou non dans ces différents “SMA”. Enfin nous avons étudié les différents moyens dont nous disposons pour appliquer les modèles provenant de la nature à la simulation.

La simulation permet de définir un degré de liberté par rapport à la réalité. Ce degré de liberté de la simulation est important lorsque la simulation se déroule dans le cadre de l'éthologie synthétique. Il est faible lorsqu'on applique la simulation à des robots réels.

Grâce au dilemme du prisonnier et aux “animats” grenouilles nous avons observé “l'agent” lorsqu'il ne communique pas. Cette étude a permis de définir le rôle et les limites de l'agent lorsqu'il est autonome face à un problème. On parle de *coordination spontanée*.

Nous avons également observé et défini deux possibilités pour communiquer dans les “SMA”. Il s'agit de l'échange de savoir qui suppose comme pour les abeilles, que les agents sont d'une part physiquement identiques et d'autre part qu'ils ont le même rôle à jouer dans leur “SMA” et qu'ils doivent communiquer. Nous avons alors parlé de systèmes multi-agents *homogènes*. De cette définition des agents homogènes découle celles des agents *hétérogènes* : ce sont des agents qui sont soit physiquement différents, soit qui ont un rôle différent dans leur “SMA”, soit les deux et qui doivent communiquer.

L'observation de fourmis ouvrières nous a montré qu'un agent peut passer par les trois stades (autonome, homogène et hétérogène) au cours de sa vie.

Pour étudier ces différents modèles de “SMA”, nous avons défini un cadre pour la simulation. C'est l'objet du chapitre 3 qui a présenté des systèmes capables d'apprendre et de réagir aux simulations provenant de l'environnement : les systèmes de classeurs.

Nous avons décrit dans ce chapitre les principaux systèmes de classeurs et leur structure. Nous avons pu voir dans un premier temps les traits communs à tous les systèmes de classeurs. Nous avons

pu ainsi dégager plusieurs caractéristiques essentielles comme la structure pyramidale de la population qui compose un système de classeurs, l'interaction nécessaire avec un environnement pour apprendre, la structure des classeurs (des règles de production composées d'une partie condition et d'une partie action) et enfin le mécanisme d'évolution directement lié à l'apprentissage résultant de l'interaction avec l'environnement. Enfin nous avons revisité les deux grandes catégories de systèmes de classeurs, avec et sans algorithmes génétiques. Les méthodes automatiques i.e. sans algorithme génétique, font actuellement l'objet d'évolutions intéressantes. Toutefois, nous nous sommes attachés à étudier les systèmes de classeurs à algorithme génétique pour leur propriété adaptative en milieu dynamique bruité. Nous avons appliqué un système purement phylogénique (les systèmes de classeurs à algorithme génétique) à un problème ontogénique : communiquer au sein des SMA.

Nous avons développé dans le chapitre 4 une version simplifiée du système de classeurs de type Pittsburgh lors d'une étude menée sur la non communication, i.e. la coordination spontanée. Nous avons étudié l'influence des principaux paramètres des systèmes de classeurs de type Pittsburgh simplifié sur le comportement des agents composant un système multi-agents se coordonnant de manière spontanée. Nous avons observé, à l'aide de mesures d'uniformisation, de capacité de mémorisation, de part de non-déterminisme et de réponse au problème posé, comment chaque agent est un individu à part entière avec un comportement qui lui est propre et qui permet au système de se coordonner. Par rapport à un tel problème, l'agent autonome composé d'un classeur de type Pittsburgh simplifié voit principalement sa précision influencée par :

- Le nombre d'individus qui améliore de manière significative la précision lorsqu'il augmente.
- Le nombre de règles qui s'il est insuffisant diminue la précision et s'il est trop grand ne l'améliore plus.
- Le temps imparti pour se coordonner qui est un facteur exogène au système.

L'inertie des agents est déterminée par :

- Le taux d'uniformisation de l'agent qui dépend lui-même de l'algorithme génétique, responsable de l'augmentation de l'inertie.
- La capacité de mémorisation qui plus elle est importante, plus l'agent restera sur ses positions et augmentera l'inertie.
- Le non-déterminisme qui régit également la capacité de l'agent à s'adapter à de nouvelles situations non apprises et qui diminue l'inertie.

Le choix, la connaissance des agents et le partage des connaissances sont des notions inhérentes au problème abordé. Les paramètres du système de classeurs ne changeront rien à ces variables. Seul un codage adapté ou une capacité de l'agent permettent d'optimiser le système. Comme nous l'avons déjà souligné, l'agent durant sa vie, peut passer de son rôle génétique à savoir l'agent autonome spécialisé dans une tâche, à un agent communicant un savoir ou encore communicant une information.

Nous avons choisi ensuite d'étudier les agents homogènes dans le chapitre 5. Les systèmes de classeurs de type Pittsburgh simplifié offrent une plate-forme adaptée à l'échange de connaissances. À partir de constats simples sur des modèles existants d'élitisme, nous avons proposé une solution

permettant d'échanger totalement la connaissance entre les agents homogènes sans pour autant que la solution ne soit trop coûteuse. Nous nous sommes inspirés de solutions améliorant localement les agents afin de trouver une solution permettant d'améliorer le système dans sa globalité. Nous y sommes parvenus en appliquant l'élitisme à la population entière des agents, créant ainsi l'élitisme distribué qui consiste à échanger à l'aide d'une réserve, les meilleurs individus des agents de manière équitable et ce en remplaçant les individus les moins performants. Cette solution a pu être appliquée car les systèmes de classeurs de type Pittsburgh simplifié ont cette particularité d'être composés d'individus qui eux-mêmes contiennent des classeurs qui participent à la solution potentielle du problème. Nous avons vu que l'élitisme distribué améliorerait notablement la performance du système en dépit d'un temps de convergence moyen plus grand que pour l'élitisme simple. C'est alors que nous avons décidé d'utiliser ces deux méthodes afin de faire progresser aussi bien localement que globalement le système multi-agents attelé à la résolution d'un problème de gestion d'une grille de 3×3 carrefours routiers en forme de croix. La solution combinée a prouvé sa supériorité et nous avons également montré sa robustesse. Nous avons obtenu une solution simple et efficace que l'on peut facilement appliquer aux systèmes multi-agents homogènes cherchant à communiquer un savoir.

Dans le chapitre 6, nous avons étudié les problèmes que peuvent engendrer l'hétérogénéité lorsque l'on cherche à échanger des informations dans un système multi-agents. Nous avons défini un modèle minimal de communication qui est composé de deux agents "jouant" au *naming games*. Le modèle minimal de communication a montré qu'il était utilisable avec un modèle ancré dans la réalité et avec deux systèmes apprenants réactifs étant capables d'utiliser une chaîne binaire : les systèmes de classeurs Pittsburgh simplifié et XCS.

Nous avons observé qu'il était nécessaire d'utiliser pleinement les ressources de communication disponibles pour que la communication soit efficace entre agents hétérogènes. Ensuite, nous avons constaté que l'homonymie et la synonymie étaient des phénomènes naturels courants qui apparaissent avec le surplus de ressources puis s'atténuent avec l'augmentation du nombre d'agents qui communiquent. Nous avons également constaté que plus le nombre d'agents est grand et plus l'usage des ressources est élevé, et plus la communication est efficace.

Enfin nous avons créé deux communautés d'agents hétérogènes qui vivent séparément dans un premier temps. Puis, nous avons imaginé trois scénarios. La *fusion* qui fait fusionner l'ensemble des deux communautés en faisant communiquer totalement les agents. Le *diplomate* est l'un des agents de chaque communauté qui va parlementer avec son homologue diplomate et enfin l'*ambassadeur* qui est la version implantée du diplomate et qui interagit en permanence avec les deux communautés. Nous avons pu observer que la fusion est la méthode la moins coûteuse pour permettre à deux communautés de se rencontrer. Toutefois cette méthode a des limites puisque lorsque la communication utilise de manière optimale les ressources disponibles, la fusion entraîne une diminution du succès de communication et fait perdre leurs origines lexicales aux communautés qui fusionnent. Le fait que la rencontre se produise relativement tôt dans l'évolution est un facteur déterminant de stabilité.

La méthode du diplomate est avantageuse dans la mesure où la communauté à laquelle il appar-

tient est suffisamment grande et où l'on utilise de manière optimale les ressources communicantes disponibles. Ce n'est encore pas la meilleure méthode pour rapprocher deux communautés mais elle permet à chacune d'entre elle de conserver le plus longtemps possible son identité.

Enfin la méthode de l'ambassadeur est un intermédiaire entre la fusion et le diplomate et préférera que l'on est trop de ressources pour bien fonctionner.

Nous avons vérifié nos hypothèses sur les trois scénarios en créant une dernière série d'expérimentations dans lesquelles le diplomate puis l'ambassadeur et enfin la fusion ont lieu après la première phase de vie séparée. Ces expérimentations ont permis de confirmer l'intérêt de chacun des scénarios présentés et d'observer que la taille du lexique est un facteur déterminant dans la réussite de la rencontre de deux mondes.

7.2 Perspectives

Cette thèse ne saurait se terminer sans parler de son devenir. Nous avons développé des solutions adaptées aux différents types de systèmes multi-agents que l'on peut rencontrer lors de la résolution de problèmes complexes. Nous avons étudié séparément des solutions pour chacune des catégories de SMA que nous avons défini. Il reste maintenant à étudier comment intégrer nos différentes solutions dans chacun des agents pour que ces agents puissent tour à tour être autonomes, homogènes et hétérogènes comme peut l'être la fourmi. Cet agent intégrant les trois solutions devra être testé et validé grâce à l'éthologie synthétique. Nous pourrons ensuite développer cette approche sur des problèmes "jouets" comme nous l'avons fait tout au long de cette thèse. Avoir un agent capable de réagir correctement à des situations multi-agents totalement différentes est une des clés permettant de résoudre les problèmes complexes.

Cette thèse nous aura permis de réfléchir plus profondément à la nature fonctionnaliste des systèmes de classeurs. Le Fonctionnalisme est une approche de l'intelligence artificielle qui prône que le cerveau humain n'est fait que de fonctions avec des entrées et des sorties. Les philosophes rétorquent aux fonctionnalistes que si le cerveau n'est que réactif alors comment se fait-il que nous ayons la notion de concepts, d'idées qui soient suffisamment abstraite pour ne pas être explicable par une simple entrée-sortie.

Nous répondons aux philosophes qu'il ne faut pas oublier le rôle de la conscience dans le fonctionnement de nos esprits [Edmonds, 2002]. En effet, celle-ci permet de stimuler en permanence différentes zones du cerveau à l'aide de situations déjà rencontrées (mémoire) et d'en simuler les réponses adaptées [Batali et Noble Grundy, 1996]. Cette phase de stimulation permet à partir de la base de connaissance existante qui provient éventuellement de l'expérience ou encore à partir de situations nouvelles imaginées, d'apprendre quels auraient été les bons choix et les bonnes réponses. "Imaginer" de nouvelles situations ou de nouvelles réponses risque de créer des préjugés lorsque la situation se produira réellement. En effet, rien ne prouve que la simulation faite par la conscience ait été correcte. Toutefois, cette capacité à stimuler toutes les zones du cerveau permet à celui-ci de continuer à

évoluer même lorsque les cinq sens ne sont pas utilisés. Il reste maintenant à essayer de comprendre la manière dont la conscience s'organise et comment elle arrive à stimuler autant de zones en même temps ? Y-a-t-il une hiérarchisation [Stone et Veloso, 2000] des stimuli, un ordre ? Si oui quel est-il ? Pourquoi sommes-nous émotif ?

Quant à l'émotion, on peut facilement imaginer qu'elle est le reflet d'une saturation de stimulations dans certaines zones du cerveau dû par exemple à la nécessité de répondre rapidement à une situation complexe. Cette saturation provient à la fois des sens réels et de la conscience. Elle entraîne une réponse parfois étrange à une situation qui, analysé *a posteriori*, était d'une relative simplicité. Pourtant le cerveau est capable d'anticiper la réponse au problème. En effet, le cerveau n'est pas obligé d'attendre que la réponse arrive au niveau "conscient" pour agir. Prenons par exemple une situation de circulation automobile banale sur une autoroute. Si une voiture fait une embardée devant votre véhicule, la peur va provoquer en vous une intense simulation de ce qu'il faut, faudrait, faudra faire pour éviter le pire. De cette émotion, vous aurez un "réflexe", une anticipation, qui est le fruit de vos préjugés par rapport à une telle situation ou le fruit de votre expérience car cela vous est déjà arrivé. Vous aurez agi avant d'avoir eu "conscience" des conséquences de vos actes grâce à cette capacité dont dispose le cerveau de continuer à apprendre des situations nouvelles même lorsqu'il ne les a jamais rencontrées.

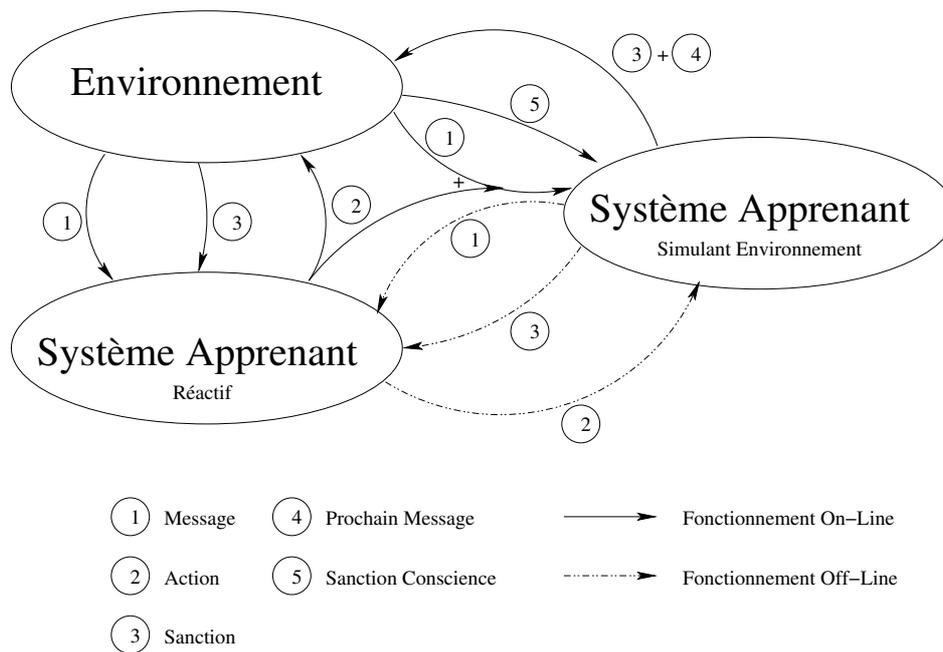


FIG. 7.1 – Modèle conscient.

Le cerveau semble d'une part fonctionnaliste avec des réactions apprises par l'expérimentation et d'autre part une conscience dont le rôle est d'apprendre de nouvelles situations à partir d'une base de connaissance existante, i.e. l'expérience, ou non pour pouvoir anticiper l'avenir. La figure 7.1 re-

présente le modèle conscient adapté à des systèmes réactifs apprenant (fonctionnalistes). Nous avons décomposé l'apprentissage en deux phases qui ont lieu pendant l'expérimentation ,i.e. c'est ce que nous nommons fonctionnement on-line. Il s'agit de phases de stimulations identiques à celles des systèmes de classeurs (cf. chapitre 3) et représentées par l'envoi d'un message par l'environnement (1) suivi d'une action du système apprenant (2) qui va servir à déterminer la sanction (3). Parallèlement, la "conscience" essaye de trouver quelles seront la sanction et le prochain message (3 + 4) de l'environnement en fonction du message envoyé (1) et de l'action (2). La conscience sera alors récompensée (5) en fonction de ses réponses. La conscience apprend donc à simuler l'environnement. Lorsque le système apprenant n'interagit plus avec l'environnement, l'ensemble du système passe en fonctionnement off-line, ou phase de sommeil, et la conscience prend le rôle de l'environnement en stimulant et récompensant le système apprenant réactif (1),(2) et (3) comme le faisait l'environnement en fonctionnement on-line.

Il serait intéressant d'appliquer ce modèle en donnant à la partie "conscience" la capacité d'être autonome, de communiquer un savoir et de communiquer des informations, tel que nous l'avons présenté dans cette thèse. On pourra alors étudier comment certains liens peuvent se créer entre les différents agents pour que le système multi-agents auxquels ils appartiennent, ait une forme de conscience globale de l'environnement qui l'entoure et par conséquent une conscience globale du problème qu'il aborde.

Bibliographie

- [Abelson et al., 2000] Harold Abelson, Don Allen, Daniel Coore, Chris Hanson, George Homsy, Thomas F. Knight, Jr., Radhika Nagpal, Erik Rauch, Gerald Jay Sussman, and Ron Weiss (2000). *Amorphous Computing*, Communications of the ACM, Vol.43, No.5, pp.74-82, 2000.
- [Aron et Passera, 2000] Aron, S. et Passera, L. (2000). *Les Sociétés Animales : Evolution de la coopération et organisation sociale*. De Boeck Université dans la série Ouvertures Psychologiques ISBN 2-8041-3442-3.
- [Arthur, 1994] Arthur, W.B. (1994). *Inductive reasoning and Bounded Rationality*. Dans American Economic Review P.P., 1994, pages 406-411.
- [Atlan, 1992] Atlan, H. (1992). *L'Organisation Biologique et la Théorie de l'Information*. Actualités scientifiques et industrielles, 1432. Hermann éditeur.
- [Baiardi et al., 1995] F. Baiardi, P. Gambini, A. Giani, A. Starita (1995). *Improving Learning through Rule Cooperation in Parallel Classifier Systems*. Présenté au Workshop on Classifier Systems, 6th International Conference on Genetic Algorithms, Pittsburgh, PA, July 1995.
- [Balch, 1997] Balch, T. (1997). *Social Entropy : a New Metric for Learning Multi-robot Teams*. Balch, T. Proc. 10th International FLAIRS Conference (FLAIRS-97), 1997.
- [Batali et Noble Grundy, 1996] Batali, J. et Noble Grundy, W. (1996). *Modeling the Evolution of Motivation*. Evolutionary Computation 4(3) :235-270, 1996.
- [Beaufils et al., 1997] B. Beaufils, J.-P. Delahaye, P. Mathieu (1997). «*Our meeting with gradual : a good strategy for the iterated prisoner's dilemma*». Dans les Proceedings of Artificial Life V, pp 202-209, ISBN 0-262-62111-8, MIT Press, 1997.
- [Bernadó, Llorà et Garell, 2001] Bernadó, E., Llorà, X. et Garell, J. M. (2001). *XCS and GALE : a Comparative Study of Two Learning Classifier Systems with*

- Six Other Learning Algorithms on Classification Tasks*. Dans les Workshop Proceedings of GECCO 2001 (San Francisco), Fourth International Workshop on Learning Classifier Systems (IWLCS 2001) pages 337-341.
- [Bonarini, 2000] Bonarini, A. (2000). *An Introduction to Learning Fuzzy Classifier Systems*. P.L. Lanzi, W. Stolzmann, et S.W. Wilson (Eds) : LCS '99, LNAI 1813, pp. 3-32, 2000. Springer-Verlag Berlin Heidelberg 2000.
- [Bonabeau et Theraulaz, 1994] Bonabeau, E. et Theraulaz, G. (1994). *Intelligence Collective*. Hermès editeur.
- [Briscoe, 1999a] Briscoe, Ted (1999). *'The acquisition of grammar in an evolving population of language agents'*. Dans Muggleton, Stephen (ed.), *Machine Intelligence, 16 : Electronic Transactions in Artificial Intelligence, Special Issue*, <http://www.etaij.org/>.
- [Briscoe E. J., 1999b] Briscoe, E.J. (1999b, submitted). *'Grammatical Acquisition : Co-evolution of Language and the Language Acquisition Device'*, *Language*, Briscoe, E.J. (1999c, in press) *'Evolutionary perspectives on diachronic syntax'*, *Proceedings of the Diachronic Generative Syntax, 5*, Oxford University Press, Oxford.
- [Bull, 1995] Bull, L. (1995). *Artificial Symbiology : Evolution in cooperative multi-agent environments*. Ph.D. Thesis, University of the West of England, Bristol.
- [Bull et al., 1995a] Bull, L., Carse, B. et Fogarty, T.C. (1995). *Evolving Multi-Agent Systems*. Dans *Genetic Algorithms in Engineering and Computer Science*, Periaux J. et Winter G., éditeurs.
- [Bull et Fogarty, 1994] Bull, L. et Fogarty, Y.C. (1994). *Evolving cooperative communicating classifier systems*. Dans Sebald, A.V. et Fogel, L.J., éditeurs. *Proceedings of the Third Annual Conference on Evolutionary Programming*, pp 308-315.
- [Bull et al., 1995b] Bull, L., T.C. Fogarty, and M. Snaith (1995). *"Evolution in Multi-Agent Systems : Evolving Communicating Classifier Systems for Gait in a Quadrupedal Robot,"*. Dans les *Proceedings of the Sixth International Conference on Genetic Algorithms*, L.J. Eshelman, (ed.). 1995, Morgan Kaufmann Publishers, Inc. : San Francisco. pp. 382-388.
- [Butz et al., 2001] Martin V. Butz, Tim Kovacs, Pier Luca Lanzi, and Stewart W. Wilson (2001). *How XCS Evolves Accurate Classifiers*. Dans les

- proceedings de GECCO-2001. Également tech. Report 2001008, University of Illinois at Urbana-Champaign, 2001.
- [Butz et Pelikan, 2001] Butz, M. et Pelikan, M. (2001). *Analyzing the Evolutionary Pressures in XCS*. Proceedings de la Genetic and Evolutionary Computation Conference 2001 (GECCO-2001) San Francisco, CA : Morgan Kaufmann. pp. 935-942
- [Butz et Wilson, 2000] Butz, M.V. et Wilson, S.W. (2000). *An Algorithmic Description of XCS*. IlliGAL Report No. 2000017.
- [Chattoe, 1998] Chattoe, E. (1998). *Just how (un)realistic are Evolutionary Algorithms as representations of social processes*. Dans le Journal of Artificial Societies and Social Simulation, 1998, vol. 1, n° 3.
- [Chavance, 2001] Chavance, B. (2001). *Organisations, institutions, système : type et niveaux de règles*. Revue d'économie industrielle, n° 97, quatrième trimestre, pp. 85-102.
- [Cheney et Seyfarth, 1982] Cheney, D.L. et Seyfarth, R.M. (1982). *How vervet monkeys perceive their grunts : Field playback experiments*. Animal Behaviour, 30, 739-751.
- [Colburn, 2000] Colburn, T.R. (2000). *Philosophy and Computer Science*. Explorations in Philosophy, James H. Fetzer, Series Editor. M.E. Sharpe. ISBN 1-56324-990-1
- [Crevier, 1993] Crevier (1993). *AI : The Tumultuous History of the Search for Artificial Intelligence*. New York : Basic Books (HarperCollins), 1993.
- [Dagaëff et Chantemargue, 1998] Dagaëff, T. et Chantemargue, F. (1998). *Reflections on Distributing agents*. Technical Report 98-19, Computer Science Department, University of Fribourg, Fribourg, Switzerland, September 1998.
- [Edwin De Jong, 1997] De Jong, E. D. (1997). *Multi-Agent Coordination by Communication of Evaluations*. Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World. (MAAMAW' 97)
- [Edwin De Jong, 1998] De Jong, E.D. (1998). *The Development of a Lexicon Based on Behavior*. Han La Poutré and Jaap van den Herik (editors) Proceedings of the Tenth Netherlands/Belgium Conference on Artificial Intelligence NAIC'98, pp. 27-36. CWI, Amsterdam, The Netherlands.

- [Edwin De Jong, 1999a] De Jong, E.D. (1999a). *Analyzing the Evolution of Communication from a Dynamical Systems Perspective*. Proceedings of the European Conference on Artificial Life ECAL'99, 689-693. ©Springer-Verlag LNCS, Berlin.
- [Edwin De Jong, 1999b] De Jong, E.D. (1999b). *Autonomous Concept Formation*. Dans les Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence IJCAI '99, pages 344-349, San Francisco, CA. Morgan Kaufmann.
- [Edwin De Jong, 2000] De Jong, E.D. (2000). *Attractors in the Development of Communication*. J.-A. Meyer, A. Berthoz, D. Floreano, H. Roitblat, and S. Wilson (Eds). SAB2000 Proceedings Supplement Book. Honolulu, Hawaii : International Society for Adaptive Behavior.
- [De Jong, 1975] De Jong, K.A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. (Thèse, Université du Michigan). Dissertation Abstracts International 36(10), 5140B. (Microfilms Universitaire No. 76-9381).
- [Delabie et al., 2000] Delabie J.H.C., Fresneau D., Pezon A., (2000). *Notes on the ecology of *Thaumatomyrmex* spp. (Hymenoptera : Formicidae : Ponerinae) in southeast Bahia, Brazil*. Sociobiology, 36, 571-584
- [Descartes, 1637] René Descartes, 1596-1650. *Discours de la méthode* (1637).
- [Dessalles, 1998a] Dessalles, J.L. (1998). *Casual conversation as logical constraint satisfaction*. Workshop on PRAGMATICS and LOGIC, ESSLLI-98 - Saarbrücken : Universität des Saarlandes. J. Allwood editeurs.
- [Dessalles, 1998b] Dessalles, J.-L. (1998). *The interplay of desire and necessity in dialogue*. Dans J. Hulstijn & A. Nijholt (ed.), Formal semantics and pragmatics of dialogue. Enschede : University of Twente, TWLT-13, 89-97.
- [Di Paolo, 1998] Di Paolo, E.A. (1998). *An investigation into the evolution of communication*. Adaptive Behavior, Vol. 6 : No. 2, 285-324.
- [Dorigo et Bersini, 1994] Dorigo, M. et Bersini, H. (1994). *A Comparison of Q-learning and classifier systems*. Dans les *Proceedings of From Animals to Animats*, Third International Conference on Simulation of Adaptive Behavior, 1994.
- [Dorigo et Schnepf, 1992] Dorigo, M. et Schnepf, U. (1992). *Genetic-based Machine Learning and Behaviour Based Robotics : A New Synthesis*. IEEE Transactions on Systems, Man and Cybernetics, Vol 22, No. 6, Novembre 1992.

- [Edmonds, 2002] Edmonds, B. (2002). *The Social Embedding of Intelligence - Towards producing a machine that could pass the Turing Test*. CPM Report n° : CPM-02-95. À paraître dans Peters, G. et Epstein, R. (eds.) *The Turing Test Sourcebook : Philosophical and Methodological Issues in the Quest for the Thinking Computer*, Kluwer.
- [Énée et Escazut, 1999] Énée, G. et Escazut, C. (1999). *Classifier Systems : Evolving Multi-Agent System with Distributed Elitism*. Dans les Proceedings du Congress on Evolutionary Computation 1999, pages 1740-1746.
- [Énée et Escazut, 2001a] Énée, G. et Escazut C. (2001a). *A Minimal Model of Communication for Multi-Agent Systems*. Dans les Proceedings de la 8th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2001) p 12-17, Antibes, FRANCE. IEEE Cat. 01TH8597. ISBN 0-7803-7241-7.
- [Énée et Escazut, 2001b] Énée, G. et Escazut, C. (2001b). *A Minimal Model of Communication for a Multi-Agent Classifier System*. Fourth International Workshop on Learning Classifier Systems (IWLCS 2001). Paru dans 2001 Genetic and Evolutionary Computation Conference Workshop Program des proceedings de GECCO 2001, p 351-356.
- [Énée et Escazut, 2002] Énée, G. et Escazut, C. (2002). *A Minimal Model of Communication for a Multi-Agent Classifier System*. (Version révisée) *Advances in Learning Classifier Systems*. LNAI 2321 (Lecture Notes in Artificial Intelligence), Pier Luca Lanzi, Wolfgang Stolzmann, Stewart W. Wilson (Eds.). Springer-Verlag Berlin Heidelberg 2002.
- [Escazut et Fogarty, 1997] Escazut, C. et Fogarty, T.C. (1997). *Coevolving Classifier Systems to Control Traffic Signals*. Dans "late breaking papers book". Genetic Programming 1997 Conference, Koza, John R. (éditeur).
- [Ferber, 1995] Ferber, J. (1995). *Les Systèmes Multi-Agents : Vers une intelligence collective*. Paru chez InterEditions dans la collection iia. ISBN : 2-7296-0665-3
- [Fogarty, Carse et Bull, 1995] Fogarty, T.C., Carse, B. & Bull, L. (1995). *Classifier Systems : selectionist reinforcement learning, fuzzy rules and communication*. Dans les Proceedings of the First International Workshop on Biologically Inspired Evolutionary Systems.
- [Forrest et Miller, 1990] Forrest, S. et Miller, J.H. (1990). *Emergent Behavior in Classifier Systems*. Dans Forrest [186], pages 213-227. ga :Forrest90b.

- [Garfield, 1985] Garfield, E. (1985). *When the Apes Speak, Linguists Listen. Part 1. The Ape Language Studies*. Dans *Essays of an Information Scientist*, Vol : 8, p. 293-301, 1985.
- [Goldberg, 1989] Goldberg, D.E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA : Addison - Wesley.
- [Harvey, 1997] I. Harvey (1997). *Is There Another New Factor in Evolution ?* Evolutionary Computation, Special Issue on Evolution, Learning, and Instinct : 100 Years of the Baldwin Effect v. 4, n. 3, pp. 311–327, 1997.
- [Hayek, 1952] Hayek, F.A. (1952). *L'ordre sensoriel, une enquête sur les fondements de la psychologie théorique*. CNRS Éditions 2001.
- [Hayek, 1992] Hayek, F.A. (1992). *Droit, législation et liberté : règles et ordre (volume 1)*. P.U.F., collection “Libre Echange”, 1992 (Troisième édition).
- [Hogg et Huberman, 1993] Hogg, T. et Huberman, B.A. (1993). *Better Than The Best : The Power of Cooperation*. L. Nadel et D. Stein eds., SFI 1992 Lectures in Complex System, 163-184, Addison-Wesley 1993.
- [Holland, 1985] Holland, J.H. (1985) *Properties of the bucket brigade algorithm*. Dans les Proceedings de la First International Conference on Genetic Algorithms and their Applications, pages 1-7, Hillsdale, New Jersey : Lawrence Erlbaum Associates.
- [Holland, 1992] Holland, J.H. (1992). *Adaptation in Natural and Artificial Systems*. A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England.
- [Holland et al., 2000] Holland, J.H., Booker, L.B., Colombetti, M., Dorigo, M., Goldberg, D.E., Forrest, S., Riolo, R.L., Smith, R.E., Lanzi, P.L., Stolzmann, W. et Wilson, S.W. (2000). *What Is a Learning Classifier System ?*. P.L. Lanzi, W. Stolzmann, et S.W. Wilson (Eds) : LCS '99, LNAI 1813, pp. 3-32, 2000. Springer-Verlag Berlin Heidelberg 2000.
- [Hume, 1750] Hume, D. (1750). *An Inquiry Concerning Human Understanding*. Republié chez C.W. Hendel, ed., *Hume : An Inquiry Concerning Human Understanding*, New York : Bobbs-Merrill, 1955.
- [Kaplan, 1999] Kaplan, F. (1999). *Dynamiques de l'Auto-Organisation Lexicale*. Dans *In Cognito*, n° 15, 1999 p 3-23.
- [Kaplan, 2000] Kaplan, F. (2000). *Semiotic schemata : Selection units for linguistic cultural evolution*. Dans Bedau, M et McCaskill, J. et Packard,

- N. et Rasmussen, S., editors, *Artificial Life VII*. The MIT Press, 2000.
- [Kaplan, 2001] Kaplan, F. (2001). *La naissance d'une langue chez les robots*. Paris, Hermès Science Publications, 2001. ISBN 2-7462-0262-X.
- [Labrou, Finin et Peng, 1999] Labrou, Y., Finin, T. et Peng, Y. (1999). *Agent Communication Languages : The Current Landscape*. IEEE Intelligent Systems, Mars / Avril 1999. 1094-7167/99.
- [Langton, 1990] Langton, C. G. (1990). *Computation at the edge of chaos : Phase transitions and emergent computation*. Elsevier Science Publishers B.V., 1990.
- [Lanzi, 1997] Lanzi, P.L. (1997). *Solving Problems in Partially Observable Environments with Classifier Systems (Experiments on Adding Memory to XCS)*. Technical Report N. 97.45
- [Lanzi et Riolo, 2000] Lanzi, P.L. et Riolo, L. (2000). *A Roadmap to the Last Decade of Learning Classifier System Research (From 1989 to 1999)*. P.L. Lanzi, W. Stolzmann, et S.W. Wilson (Eds) : LCS '99, LNAI 1813, pp. 3-32, 2000. Springer-Verlag Berlin Heidelberg 2000.
- [Lanzi et Wilson, 2000] Lanzi, P.L. et Wilson, S.W. (2000). *Toward Optimal Classifier System Performance in Non-Markov Environments*.
- [Littman, 1994] Littman, M.L. (1994). *Memoryless policies : theoretical limitations and practical results*. Dans D. Cliff, P. Husbands, J.-A. Meyer, et S.W. Wilson (Eds.). *From Animals to Animat 3 : Proceedings de la Third International Conference on Simulation of Adaptive Behavior*. Cambridge, MA : MIT Press/Bradford Books.
- [MacLennan, 1991] MacLennan, B.J. (1991). *Synthetic ethology : An approach to the study of communication*. Dans les Proceedings of the 2nd interdisciplinary workshop on synthesis and simulation of living systems, pages 631–658, 1991.
- [MacLennan, 1999] MacLennan, B.J. (1999). *The Emergence of Communication through Synthetic Evolution*. Technical Report UT-CS-99-431, October 20, 1999. Paru dans *Advances in Evolutionary Synthesis of Neural Systems*, edited by Vasant Honavar, Mukesh Patel, and Karthik Balakrishnan (MIT Press).
- [MacLennan et Burghardt, 1994] MacLennan, B.J. et Burghardt, G.M. (1994). *Synthetic Ethology and the Evolution of Cooperative Communication*. Dans *Adap-*

- tive Behavior Vol. 2, No. 2, 161-188. Massachusetts Institute of Technology.
- [Mataric, 1996] Mataric, M. (1996). *Using Communication to Reduce Locality in Distributed Multi-agent learning*. Brandeis University Computer Science Technical Report CS-96-190, Nov 1996.
- [Maturana, Shen et Norrie, 1999] Maturana, F., Shen, W. et Norrie, D. H. (1999). *Metamorph : An Adaptive Agent-Based Architecture for Intelligent Manufacturing*. International Journal of Production Research, 37(10), 2159-2174.
- [Michel, 1995] Michel, O. (1995) Khepera simulator homepage : <http://www.epfl.ch/lami/team/michel/khep-sim/>
- [Mikami et Kakazu, 1993] Mikami, S. et Kakazu, Y. (1993). *Self-organized control of traffic signals through genetic reinforcement learning*. Dans Proceedings of the IEEE Intelligent Vehicles Symposium, pp 113-118.
- [Mikami et Kakazu, 1994] Mikami, S. et Kakazu, Y. (1994). *Genetic Reinforcement Learning for Cooperative Traffic Signal Control*. Dans les Proceedings of The First IEEE Conference on Evolutionary Computation : 223-228 (1994).
- [Miller et Butts, 1998] Miller, J.H. et Butts, C. (1998). *Communication and Cooperation*. Journal of Economic Behavior and Organization 47 :2 [2002]. Revised. April 24, 1998.
- [Montana et Czerwinski, 1996] Montana, D.J. et Czerwinski, S. (1996). *Evolving Control Laws for a Network of Traffic Signals*. Proceedings de la première conférence annuelle de Genetic Programming 1996. J.R. Koza, D.E. Goldberg, D. Fogel et R.L. Riolo éditeurs. MIT Press.
- [Noble,1998] Noble, J. (1998). *The Evolution of Animal Communication Systems : Questions of Function Examined through Simulation*. Ph.D. Thesis in University of Sussex.
- [Noble et Cliff, 1996] Noble, J., et Cliff, D. (1996). *On simulating the evolution of communication*. Dans P. Maes, M. Mataric, J.-A. Meyer, J. Pollack et S. W. Wilson (Eds.), From animals to animats 4 : Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, pp. 608-617. Cambridge (MA) : MIT Press.
- [Pasteels et al., 1987] Pasteels, J.M., Deneubourg, J. et Goss, S. (1987). *Self-organization mechanisms in ant societies : Trail recruitment to newly discovered food sources*. In Pasteels, J.M. et Deneubourg,

- J. (eds), From individual to collective behavior in social insects. Basel : Birkäuser Verlag, 1987.
- [Platon] Platon. *Alcibiades I*, partiellement réimprimé dans A. Flew, ed., *Body, Mind, and Death*, 1972, 34-71.
- [Quinn, 2000] Quinn, M. (2000). *Evolving Co-operative Homogeneous Multi-Robot Teams*. Proceedings de la IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000). Takamatsu, Japan.
- [Reinhard et Kaib, 2001] Reinhard, J. et Kaib, M. (2001). *Trail Communication During Foraging and Recruitment in the Subterranean Termite *Reticulitermes santonensis* De Feytaud (Isoptera, Rhinotermitidae)*. In Journal of Insect Behavior, Vol. 14, No. 2, 2001. Plenum Publishing Corporation.
- [Rosin et Belew, 1997] Rosin, C.D. et Belew, R.K. (1997). *New Methods for Competitive Coevolution*. Evolutionary Computation 5(1) : 1-29, The MIT press 1997.
- [Rustogi et Singh, 1999] Rustogi, S.K. et Singh, M.P. (1999). *Be Patient and Tolerate Imprecision : How Autonomous Agents can Coordinate Effectively*. Dans les Proceedings de IJCAI-99, p. 512-519.
- [Samuels et al., 2001] Samuels, Stich et Tremoulet (2001). *Rethinkink Rationality : From Bleak Implications to Darwinian Modules*. À paraître dans Rutgers University Invitation to Cognitive Science, E. LePore et Z. Pylyshyn, éditeurs.
- [Sen et Sekaran, 1995] Sandip Sen et Mahendra Sekaran (1995). *"Multiagent coordination with learning classifier systems"*. Dans les Working Notes of the IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems (pages 84-89), Montreal, Canada, August 1995.
- [Seyfarth, Cheney et Marler, 1980] Seyfarth, R., Cheney, D.L., et Marler, P. (1980). *Monkey responses to three different alarm calls : Evidence of predator classification and semantic communication*. Science, 210, 801-803.
- [Shannon et Weaver, 1949] Shannon, C.E. et W. Weaver (1949). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, Ill.
- [Sigaud et Gérard, 2000] Sigaud, O. et Gérard, P. (2000). *The use of roles in a multiagent adaptive simulation*. Dans les Proceedings of the 14th European Conference in Artificial Intelligence, Workshop on Balancing reactivity and Social Deliberation in Multiagent Systems.

- [Smith, 1980] Smith, S. (1980). *A learning system based on genetic algorithms*. PhD. Thesis, Computer Science Department, University of Pittsburgh, Decembre 1980.
- [Smith, 1984] Smith, S.F. (1984). *"Adaptive Learning Systems"*. Dans Expert Systems : Principles and Case Studies, (ed.) R. Forsyth, Associated Book Publishers Ltd, October, 1984.
- [Spiessens, 1990] Spiessens, P. (1990). *PCS : A classifier system that builds a predictive internal world model*. Dans Luigia Carlucci Aiello, editor, ECAI 90 9th European Conference on Artificial Intelligence, pages 622–627, Stockholm, 6.-10. August 1990. Pitman Publishing, London.
- [Steels, 1996a] Steels, L. (1996). *A self-organizing spatial vocabulary*. Artificial Life Journal. 2(3) 1996.
- [Steels, 1996b] Steels, L. (1996). *Self-organising vocabularies*. Proceedings de Artificial Life V, 1996, Nara, Japon, Langton C. editeur.
- [Steels, 1996c] Steels, L. (1996). *The spontaneous Self-organization of an Adaptive Language*. Dans Furukawa, K., D. Michie et S. Muggleton (eds.) (1996) Machine Intelligence 15. Oxford University Press, Oxford. pp. 205-224.
- [Steels, 1997] Steels, L. (1997). *The synthetic modeling of language origins*. Evolution of Communication Journal John Benjamins, Amsterdam.
- [Steels et Kaplan, 1999] Steels, L. et Kaplan, F. (1999). *Situated Grounded Word Semantics*. Dans Dean, T. (ed) Proceedings of IJCAI 99, Morgan Kaufmann Publishers, San Francisco, CA, p.862-867.
- [Steels et McIntyre, 1999] Steels, L. et A. McIntyre (1999). *Spatially Distributed Naming Games*. Journal of Complex Systems, Hermes Pub., Paris.
- [Stefens et Merx, 1990] Stefens, L.M. et Merx, M.B. (1990). *The effect of agent control strategy on the performance of a dai pursuit problem*. In Proceedings of the 1990 Distributed AI Workshop.
- [Stolzmann, 1998] Stolzmann, W. (1998). *Anticipatory Classifier Systems*. Dans Koza, John R., Banzhaf, Wolfgang, Chellapilla, Kumar, Deb, Kalyanmoy Dorigo, Marco, Fogel, David B., Garzon, Max H., Goldberg, David E., Iba, Hitoshi, et Riolo, Rick. (editors). Genetic Programming 1998 : Proceedings of the Third Annual Conference, July 22-25, 1998, University of Wisconsin, Madison, Wisconsin, 658-664. San Francisco, CA : Morgan Kaufmann.

- [Stone et Veloso, 2000] Stone, P. et Veloso, M. (2000). *Layered Learning*. Eleventh European Conference on Machine Learning (ECML-2000).
- [Tan, 1993] Tan, M. (1993), *Multi-Agent Reinforcement Learning : Independent vs. Cooperative Agents*. Dans 'Proceedings, Tenth International Conference on Machine Learning ', Amherst, MA, pp. 330–337.
- [Turney, 1996] Turney, P. (1996). *How to Shift Bias : Lessons from the Baldwin Effect*. *Evolutionary Computation*, 4 (3), 271–295
- [van Eijk et al., 1998] van Eijk, R.M., de Boer, F.S., van der Hoek, W. et Meyer, J.-J.Ch. (1998). *Systems of Communicating Agents*. Dans Henri Prade, editor, *Proceedings of the 13th biennial European Conference on Artificial Intelligence (ECAI'98)*, pages 293–297. John Wiley & Sons, Ltd, 1998.
- [Valenzuela-Rendón, 1998a] Valenzuela-Rendón, M. (1998a). *Reinforcement learning in the fuzzy classifier system*. *Expert Systems with Applications*, 14, 237-247.
- [von Frisch, 1967] von Frisch, K. (1967). *The Dance Language and Orientation of Bees*. The Belknap Press of Harvard University Press, Cambridge, Mass.
- [Wagner, 1999] Wagner, K. (1999). *Habitat, communication and cooperative strategies*. Dans Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., et Smith, R. E. (eds.). *GECCO-99 : Proceedings of the Genetic and Evolutionary Computation Conference, July 13-17, 1999, Orlando, Florida USA*. San Francisco, CA : Morgan Kaufmann, pp. 694-701.
- [Watkins, 1989] Watkins, C.J.C.H. (1989). *Learning from Delayed Rewards*. Ph.D. thesis, King's College, Cambridge University, 1989.
- [Werner et Dyer, 1991] Werner, G. M. et Dyer, M. G. (1991). *Evolution of Communication in Artificial Organisms*. *Artificial Life II, SFI Studies in the Sciences of Complexity*, vol. X, édité par C.G. Langton, C. Taylor, J.D. Farmer et S. Rasmussen. Addison-Wesley, 1991.
- [Wilson, 1987] Wilson, S.W. (1987). *Classifier Systems and the Animat Problem*. *Machine Learning 2* : 199-228, 1987. Kluwer Academic Publishers, Boston.
- [Wilson, 1994] Wilson, S. W. (1994). *ZCS, a zeroeth level classifier system*. *Evolutionary Computation*. Vol. 2, no.1, pp. 1-18.

- [Wilson, 1995] Wilson, S.W. (1995). *Classifier fitness based on accuracy*. Evolutionary Computation, 3(2), pp 149-175.
- [Wilson, 1998] Wilson, S.W. (1998). *Generalization in the XCS Classifier System*. Dans Genetic Programming 1998 : Proceedings de la Third Annual Conference, J. Koza et al. (Eds.) San Francisco, CA : Morgan Kaufmann.
- [Xuan et al., 2001] Xuan, P., Lesser, V., Zilberstein, S. (2001). *Communication Decisions in Multi-agent Cooperation : Model and Experiments*. Dans AGENTS'01, du 28 Mai au 1 Juin 2001, Montréal, Quebec, Canada.