

ACADÉMIE DE MONTPELLIER

UNIVERSITÉ MONTPELLIER II

— SCIENCES ET TECHNIQUE DU LANGUEDOC —

THÈSE

présentée à l'Université Montpellier II - Sciences et Techniques du Languedoc  
pour obtenir le diplôme de DOCTORAT

SPÉCIALITÉ : **INFORMATIQUE**  
*Formation Doctorale* : **Informatique**  
*Ecole Doctorale* : **Information, Structures et Systèmes**

**Le modèle satisfaction-altruisme :  
coopération et résolution de conflits entre  
agents situés réactifs, application à la robotique**

par

Olivier SIMONIN

Soutenu le 20 Décembre 2001 devant le Jury composé de :

M. Alexis DROGOUL, Maître de Conférence Hdr., LIP6 UPMC Paris, ..... Rapporteur  
M. Dominique DUHAUT, Professeur, Université Bretagne Sud Lorient, ..... Rapporteur  
M. Rachid ALAMI, Directeur de Recherche, CNRS du LAAS Toulouse, ..... Examineur  
M. Jean-Claude KONIG, Professeur, LIRMM UM2 Montpellier, ..... Président  
M. Joël QUINQUETON, Professeur, LIRMM UM3 Montpellier, ..... Examineur  
M. Jacques FERBER, Professeur, LIRMM UM2 Montpellier, ..... Directeur de Thèse  
M. Alain LIÉGEOIS, Professeur, LIRMM UM2 Montpellier, ..... Co-Directeur de Thèse

et sur avis du rapport de :

M. Luc STEELS, Professeur, Université Libre de Bruxelles, ..... Rapporteur



## Remerciements

Je tiens à exprimer ma profonde reconnaissance à mes deux directeurs de thèse, qui m'ont accompagné et soutenu tout au long de ces trois belles années.

Je voudrais tout d'abord remercier Jacques Ferber pour m'avoir transmis sa passion de l'intelligence artificielle et sa vision de l'aventure scientifique. J'ai apprécié tout au long de ces années les échanges, autant scientifiques qu'amicaux, qui ont caractérisé notre collaboration. Je lui suis très reconnaissant pour la confiance et la liberté qu'il m'a accordées depuis mes débuts en DEA, mais aussi pour tous les conseils précieux qu'il a su me communiquer. Ils m'ont permis d'éviter les pièges et d'arriver à bon port, merci !

Ma rencontre avec Alain Liégeois a été tout aussi importante et décisive dans l'évolution de mon travail. Je lui suis très reconnaissant pour m'avoir initié aux problèmes de la robotique et pour m'avoir donné la possibilité d'expérimenter mes travaux théoriques sur de véritables robots. Je n'oublierai pas non plus sa rigueur scientifique et l'impact qu'elle a pu avoir sur l'ensemble de mon travail. Enfin, je voudrais dire combien la dynamique et la bonne ambiance régnant au sein de son équipe ont été pour moi une formidable source de motivation.

Je tiens à exprimer ma gratitude à Alexis Drogoul pour le travail exceptionnel qu'il a réalisé en rédigeant un long et précieux rapport. Cette lecture fine du manuscrit est pour moi une source de réflexions et de projets sans fin. Je suis d'autant plus honoré de l'intérêt qu'il a porté à mon travail que sa thèse a été durant ces dernières années mon ouvrage de référence.

Mes remerciements vont également à Dominique Duhaut pour avoir accepté d'être rapporteur de ce travail. Les remarquables discussions que nous avons pu avoir sur les choix technologiques que j'ai proposé me permettent d'envisager de plus riches perspectives.

Je tiens à remercier Luc Steels d'avoir pu m'accorder une partie de son temps, que je sais précieux, pour établir un rapport supplémentaire détaillé et constructif.

Mes remerciements les plus vifs à Jean-Claude König pour avoir accepté de tenir le rôle de président, mais aussi pour m'avoir écouté et conseillé lorsque je me suis aventuré, en troisième année, dans l'univers complexe des réseaux de communication.

Je tiens également à remercier Rachid Alami et Joël Quinqueton pour avoir accepté de participer au jury et de commenter mes travaux. J'espère que nos discussions pourront se poursuivre prochainement au détour d'une nouvelle conférence...

Bien sûr, je tiens à remercier chaleureusement Philippe Lucidarme, notre créateur de robots. Sans lui, nous n'aurions certainement pas pu expérimenter si rapidement et si efficacement nos idées. Je le remercie encore pour sa patience et sa pédagogie, et j'espère que nous pourrions continuer longtemps cette fructueuse coopération.

Cette thèse n'aurait pas pu se passer dans d'aussi bonnes conditions sans la présence précieuse de mes compagnons d'équipe (un vrai SMA !). Je pense en particulier à Jérôme Chapelle (merci pour cette belle collaboration et ta disponibilité, je te souhaite beaucoup de succès), à Marc Ricordeau (pour nos discussions scientifiques et géopolitiques sans fins !),

à Fabien Michel (merci pour tes relectures d'articles, à toi la thèse maintenant !), à Pierre Bommel (reste aussi passionné et passionnant), à Olivier Gutknecht (Monsieur Madkit !) et à Vincent Decugis (mon prédécesseur ?).

Un grand merci à Pascal Sitbon, pour son soutien inconditionnel, sa relecture si "pro" du chapitre consacré au protocole de communication, et bien sûr, pour les phases de décompressions si nombreuses et si nécessaires durant la thèse !

Je tiens à remercier tout autant celles et ceux qui m'ont donné les coups de mains les plus importants, c'est-à-dire ceux des derniers jours et des dernières heures. En particulier je pense à la formidable Maud Giorgi (des achats de l'apéro au soutien moral depuis la Maîtrise, et puis tout le reste...), bien sûr au grand Eric Thierry (l'exemple à suivre, sauf pour les nuits blanches de boulot !) et enfin à l'incroyable Alexis Irlande (des photocopies de la thèse aux dépannages réseaux, système et de la voiture !).

Un immense merci à Anna et Vikki pour leur aide dans mon pénible apprentissage de la langue de Shakspeare, mais surtout pour votre bonne humeur inaltérable et votre amitié sans faille.

Un grand merci à tous les thésards de robotique, pour votre soutien et votre sympathie lors de mes nombreuses descentes au rez de chaussée (jamais en enfer !). Je voudrais ajouter ici un remerciement spécial à Olivier Strauss pour m'avoir accueilli (et supporté) dans son bureau, moi et mes robots, durant quelques trois semaines.

Je tiens à remercier les ingénieurs et directeurs de la société Wany, et en particulier leur gourou Erwann Lavarec, qui nous ont apportés leur aide technique et matérielle lors de la réalisation du premier prototype de robot mobile. C'était d'autant plus sympa que Wany vivait alors ses premières heures. Encore une fois, merci, et longue vie à vous...

Je voudrais remercier les (ex)thésards qui ont partagé un bureau, un article ou une réflexion durant ces trois années. Je pense en particulier à Stéphane Coulondre, Philippe Rongier, Frédéric Koriche, Youssef Hamadi, Guillaume Damiand, Jean-François Baget, Florent Masegla, Xavier Baril, Nicolas Vidot, Mohamed Bouklit, Fabien de Montgolfier, Frédéric Souchon et Stéphanie Bocquet.

Enfin, je voudrais remercier tous ceux qui à un moment donné ou à un autre, volontairement ou pas, m'ont fait avancer vers l'aboutissement de cette thèse. Je pense à Philou, Eliza, Myriam, Olivier R., Delphine, Christine, Christophe F., Gab., Françoise, Phédia, Stéphan C., Jean-Marc L., Audrey D., Antony, Frank, Laurence, Cyrus, Madli... et pardon à tous ceux que j'oublie !

Pour terminer, je tiens à remercier mes plus grands supporters que sont ma soeur Magali et mes frères Frédéric et Gilles, et bien sûr,

mes parents, à qui je dédie cette thèse.

# Résumé

Cette thèse se situe à la rencontre du domaine des systèmes multi-agents (SMA) et de la robotique mobile. Le modèle d'agent développé permet d'introduire des comportements coopératifs intentionnels dans l'approche de résolution collective de problèmes.

Après avoir analysé les situations/actions coopératives entre agents situés, nous définissons deux états de satisfaction évalués par l'agent. D'une part, la satisfaction personnelle, qui est une mesure de la progression des actions de l'agent. D'autre part, la satisfaction interactive qui évalue les actions de son voisinage : gêne, aide (potentielle), indifférence. Les agents peuvent alors s'influencer en émettant des signaux répulsifs ou attractifs. En réception, si ces influences sont plus intenses que la satisfaction personnelle, elles déclenchent des réactions altruistes.

L'architecture bâtie sur ce principe est évaluée par le traitement, en simulation, de problèmes physiquement distribués. Le problème des robots fourrageurs et celui des conflits spatiaux sont résolus efficacement par notre modèle, combinant auto-organisation et comportements coopératifs intentionnels. Le modèle est ensuite appliqué à un système hétérogène, puis il est étendu par l'introduction d'un module d'apprentissage exploitant les mesures de satisfactions.

Dans le cadre d'une implémentation réelle, nous proposons un protocole de communication générique dédié aux SMAs situés. Celui-ci établit, par une méthode de synchronisation indirecte, un multiplexage temporel assurant une fréquence de communication constante aux agents, et ce, quel que soit leur nombre. Enfin, le modèle proposé, ainsi que le protocole de communication, sont validés expérimentalement par la résolution de conflits spatiaux entre robots mobiles autonomes.



# Table des matières

<b>I</b>	<b>Problématique</b>	<b>9</b>
<b>1</b>	<b>Introduction</b>	<b>11</b>
1.1	Cadre général . . . . .	11
1.2	Combiner coopération directe et indirecte... . . . .	15
1.3	Plan de la thèse . . . . .	17
<b>2</b>	<b>Les systèmes multi-agents situés</b>	<b>19</b>
2.1	Les systèmes multi-agents . . . . .	19
2.1.1	Les différents types de systèmes multi-agents . . . . .	20
2.1.2	Objectifs et problèmes du domaine . . . . .	23
2.2	Des agents situés . . . . .	24
2.3	L'approche réactive . . . . .	25
<b>3</b>	<b>Le problème de la coopération</b>	<b>29</b>
3.1	Problème général . . . . .	29
3.2	Techniques de coopération . . . . .	30
3.2.1	Définir la coopération . . . . .	30
3.2.2	Les méthodes de coopération . . . . .	31
3.2.3	Bilan . . . . .	34
3.3	Principe de notre approche . . . . .	35
3.3.1	Actions coopératives . . . . .	35
3.3.2	Coordination réactive et actions coopératives . . . . .	37
3.4	Organisation des chapitres suivants . . . . .	38
<b>II</b>	<b>Modélisation des satisfactions</b>	<b>41</b>
<b>4</b>	<b>Modélisation de la satisfaction personnelle d'un agent situé</b>	<b>43</b>

4.1	Introduction . . . . .	43
4.2	Mesures de la progression des tâches . . . . .	46
4.2.1	Tâches personnelles . . . . .	46
4.2.2	Notion d'exécution de tâche . . . . .	46
4.2.3	Mesures classiques de la progression des tâches . . . . .	48
4.3	Satisfaction personnelle . . . . .	50
<b>5</b>	<b>Signaux de satisfaction interactive et réaction altruiste</b>	<b>57</b>
5.1	Objectif . . . . .	57
5.1.1	Évaluer les situations nécessitant des actions coopératives . . . . .	59
5.1.2	Un mécanisme de communication et de réaction temps réel . . . . .	60
5.2	Signaux de satisfaction interactive . . . . .	61
5.3	Réaction altruiste . . . . .	62
5.3.1	Nature des signaux . . . . .	64
5.4	vers la satisfaction du système... . . . .	66
<b>III</b>	<b>Le modèle Satisfaction-Altruisme</b>	<b>67</b>
<b>6</b>	<b>Coopération par altruisme et signaux de satisfaction</b>	<b>69</b>
6.1	Comportements altruistes . . . . .	69
6.2	Principe de coopération . . . . .	70
6.3	Des interactions satisfaisantes . . . . .	71
6.4	Le modèle satisfaction-altruisme . . . . .	74
6.4.1	Sélection d'une action ou tâche (en temps réel) . . . . .	74
6.4.2	Étude et analyse des systèmes à base de champs de potentiels (forces)	76
6.4.3	Intégration de la sélection de tâche et les actions coopératives au niveau vectoriel . . . . .	80
6.4.4	Choix du signal à émettre . . . . .	84
6.4.5	Le modèle général . . . . .	85
<b>7</b>	<b>Architecture et Algorithmes</b>	<b>87</b>
7.1	Sélection d'une fonction . . . . .	87
7.2	Le noyau du modèle (sélection et combinaison) . . . . .	93
7.2.1	Algorithme . . . . .	93
7.2.2	Un processus temps réel . . . . .	94
7.3	De l'interaction à la propagation d'informations . . . . .	96

7.3.1	Module d'émission et de propagation des signaux . . . . .	99
7.4	Comparaison à d'autres architectures . . . . .	100
7.4.1	ALLIANCE (L. Parker) . . . . .	100
7.4.2	Une architecture coopérative et émotionnelle . . . . .	101
7.5	Organisation des chapitres suivants . . . . .	103

## IV Application à la résolution de problèmes distribués et Simulations 105

<b>8</b>	<b>Des robots fourrageurs</b>	<b>107</b>
8.1	Présentation du problème . . . . .	107
8.1.1	Introduction . . . . .	107
8.1.2	Des robots footballeurs? . . . . .	109
8.1.3	Travaux liés aux robots fourrageurs . . . . .	110
8.2	Application de l'architecture aux robots fourrageurs . . . . .	111
8.2.1	Traitement distribué du problème . . . . .	111
8.2.2	Vers des robots fourrageurs altruistes . . . . .	116
8.3	Simulations, résultats et analyses . . . . .	117
8.3.1	Les simulateurs . . . . .	117
8.3.2	Implémentation de l'algorithme de référence . . . . .	118
8.3.3	Résultats et analyse du modèle sans signaux . . . . .	122
8.3.4	Modèle AVEC SIGNAUX (SANS propagation) . . . . .	124
8.3.5	Modèle AVEC propagation des attractions . . . . .	130
8.3.6	Variation dynamique des portées d'émissions . . . . .	132
8.4	Bilan . . . . .	134
<b>9</b>	<b>Traitement des conflits spatiaux</b>	<b>137</b>
9.1	Introduction . . . . .	137
9.2	Situations conflictuelles . . . . .	138
9.3	Principe de résolution . . . . .	140
9.4	Eco-robots et règles comportementales . . . . .	141
9.4.1	Une application de l'éco-résolution à des robots mobiles . . . . .	142
9.4.2	Règles de résolution . . . . .	143
9.5	Applications . . . . .	147
9.5.1	Piéger les robots . . . . .	147
9.5.2	Étude, Simulations et analyses . . . . .	149

9.6	Eco-résolution et modèle satisfaction-altruisme . . . . .	156
9.7	Bilan . . . . .	157
<b>10</b>	<b>Système hétérogène et introduction de l'apprentissage</b>	<b>159</b>
10.1	Introduction . . . . .	159
10.1.1	Hétérogénéité des robots et des objets . . . . .	159
10.2	Simulations et comportements coopératifs . . . . .	162
10.2.1	Paramètres . . . . .	162
10.2.2	Simulations . . . . .	163
10.3	Introduction de l'apprentissage . . . . .	165
10.3.1	Éviter les conflits, apprendre à coopérer . . . . .	165
10.3.2	Intégration de l'apprentissage dans le modèle . . . . .	166
10.3.3	Simulations et analyse des résultats . . . . .	169
10.4	Bilan . . . . .	174
<b>V</b>	<b>Communications et expérimentations en robotique mobile distribuée</b>	<b>177</b>
<b>11</b>	<b>Un protocole de communication pour SMA situé</b>	<b>179</b>
11.1	Le problème . . . . .	179
11.1.1	Choix technologiques . . . . .	180
11.1.2	Caractéristiques du protocole recherché . . . . .	181
11.2	Les protocoles existants . . . . .	182
11.2.1	L'approche supervisée . . . . .	182
11.2.2	Solutions distribuées . . . . .	183
11.3	Proposition d'un protocole dynamique . . . . .	184
11.3.1	Spécificités d'un réseau de robots mobiles autonomes . . . . .	184
11.3.2	Proposition d'un protocole par allocation dynamique et synchronisation indirecte . . . . .	186
11.3.3	Résolution des collisions et convergence . . . . .	189
11.3.4	Paramètres et efficacité . . . . .	191
11.4	Simulations et optimisation de la probabilité de réallocation . . . . .	195
11.5	Bilan sur le protocole proposé . . . . .	198
<b>12</b>	<b>Validation sur des mini-robots mobiles</b>	<b>201</b>
12.1	Introduction . . . . .	201

## TABLE DES MATIÈRES

7

12.2	Navigation autonome et comportements réactifs . . . . .	202
12.2.1	Implémentation des communications et des perceptions . . . . .	204
12.2.2	Navigation et comportement . . . . .	209
12.3	Expérimentation du modèle satisfaction-altruisme . . . . .	212
12.3.1	Programmation du modèle Satisfaction-Altruisme sur les robots . .	212
12.3.2	Résolution du problème du corridor fermé . . . . .	213
12.4	Bilan . . . . .	216
<b>VI</b>	<b>Conclusion et Annexes</b>	<b>219</b>
<b>13</b>	<b>Conclusion</b>	<b>221</b>
13.1	Bilan . . . . .	221
13.2	Perspectives . . . . .	224
<b>14</b>	<b>Annexes</b>	<b>227</b>
14.1	Simulations . . . . .	227
14.1.1	Robots fourrageurs . . . . .	227
14.1.2	Calcul de trajectoire des éco-robots . . . . .	227
14.2	Résolution des conflits spatiaux . . . . .	228
14.2.1	Preuve de résolution pour le problème de l'impasse . . . . .	228
14.3	Simulation du protocole de communication . . . . .	230



Première partie

Problématique



# Chapitre 1

## Introduction

### 1.1 Cadre général

Cette thèse se situe à la rencontre de deux domaines de recherche : l'intelligence artificielle distribuée (IAD) et la robotique mobile. Notre étude concerne le problème de la *coopération* entre plusieurs entités autonomes (ou agents), c'est-à-dire la recherche d'interactions intelligentes permettant aux agents d'effectuer efficacement leurs tâches. C'est un problème central dans le domaine des *systèmes multi-agents (SMA) réels ou simulés*.

Historiquement, on peut dire que l'approche "multi-agents" a émergée de celle de l'IA vers le milieu des années 80. Elle concerne l'étude et la conception de systèmes *distribués* faisant *interagir plusieurs agents non supervisés*. Il s'agit bien d'un changement radical dans l'approche du traitement des problèmes. La vision mono-agent ou celle d'un système réparti supervisé, trop monolithique et fragile, est rejetée. Cette approche a plusieurs sources : déception des résultats de l'IA classique, développement de l'algorithmique distribuée et des réseaux, étude des systèmes vivants par des chercheurs en informatique et retour en force de la vision systémique [de Rosnay, 1977].

Du côté de la robotique, à la même période, apparaît le besoin de concevoir des systèmes robotisés plus robustes et capables d'une plus grande adaptabilité face aux variations de l'environnement. Les champs d'applications principaux étant l'exploration spatiale, l'exploration sous-marine, la récupération de déchets toxiques, les grands chantiers, l'intervention en milieux forestier, radio-actifs, de catastrophes naturelles et les opérations de déminage. Ce sont toutes des applications en milieux dangereux pour l'homme ou difficiles d'accès. Or, si l'homme est absent du terrain d'action, les robots doivent être capable d'agir *seuls* et de *s'adapter* à un environnement mal connu et/ou changeant.

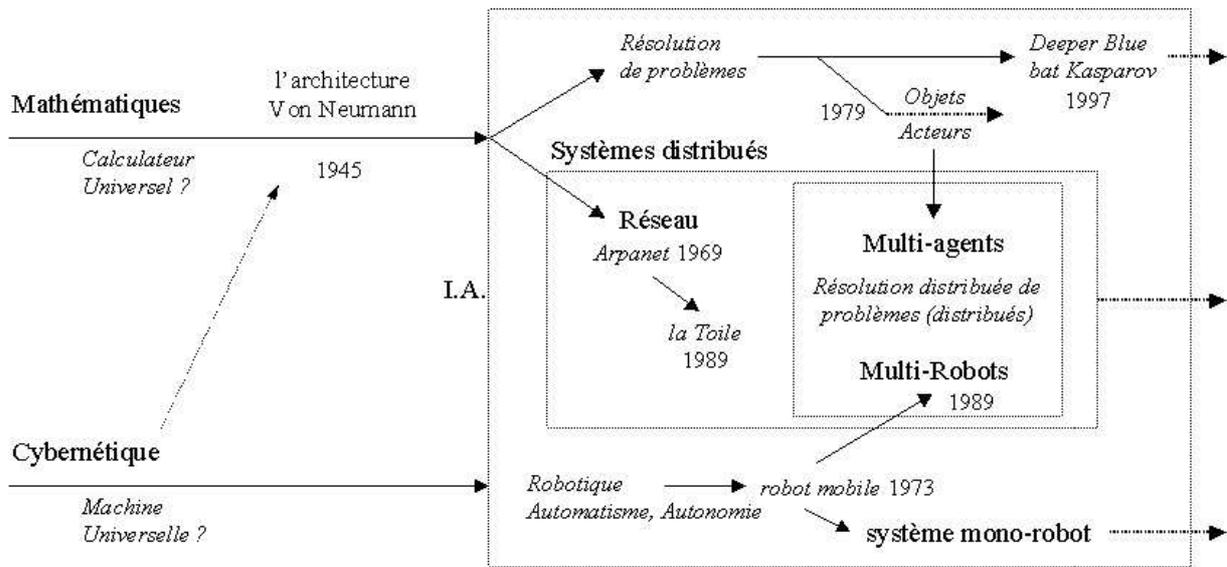


FIG. 1.1 – Schéma de l'évolution de l'informatique et de la robotique vers les systèmes distribués

L'approche multi-agents, qui apparaît sous l'essor de la programmation par objets, du parallélisme et des réseaux de communication (cf. figure 1.1), s'intéresse au fonctionnement des systèmes distribués à la fois au niveau global, au niveau des interactions et au niveau de chaque agent.

Cette nouvelle vision informatique des systèmes, des problèmes et de leur résolution couvre un champ d'études très vaste. Les grands domaines d'application sont la *résolution de problèmes* (la résolution distribuée de problèmes, la résolution de problèmes distribués<sup>1</sup> et les techniques distribuées pour la résolution de problèmes), la *simulation multi-agent* (vie artificielle, éco-systèmes, théorie des jeux), la *robotique distribuée* (robotique mobile et robotique cellulaire) et les *agents communicants (du Web)*.

Mais au delà de ces grands domaines d'applications, les systèmes multi-agents traitent de problèmes classiques en IAD :

- comment concevoir une entité artificielle capable de *percevoir* et d'*agir* sur son environnement ?
- comment faire interagir, *coopérer-coordonner-concurrer*, des agents autonomes évoluant dans le même univers ?
- comment leur faire *résoudre* des *problèmes distribués* et/ou *dynamiques* ?
- comment les faire *communiquer* ?

<sup>1</sup>nous parlerons de problème distribué lorsque celui-ci est fonctionnellement ou physiquement distribué.

## Le problème de la coopération

Notre étude se concentre sur le problème très général de la *coopération* entre agents situés (ou robots) autonomes. Il s'agit d'intégrer cette notion au coeur des agents participant à la résolution de problèmes distribués. En effet, la coopération est parfois nécessaire à l'accomplissement de certaines tâches et peut être synonyme d'accroissement des performances :

- réalisation d'une tâche qui requiert la combinaison quantitative des efforts de plusieurs agents,
- réalisation d'une tâche qui requiert la combinaison de compétences diverses réparties chez divers agents spécialisés (notion qualitative),
- évitement ou résolution de conflits (liés à l'utilisation d'une ressource),
- entre-aide, répartition des tâches et partage des ressources pour l'amélioration des performances individuelles ou globales.

La coopération est une des caractéristiques les plus intéressantes des systèmes multi-agents. Elle est d'ailleurs à la source même du développement du domaine. En effet, un système n'est performant que si les agents sont capables de s'organiser pour produire un tout supérieur à la somme des performances individuelles.

Les méthodes ou techniques de coopération proposées depuis une quinzaine d'années ont naturellement suivi l'évolution de l'IA(D).

Les premiers systèmes coopératifs ont été conçus sur la base de communications de haut niveau. Les échanges de messages sont effectués par des agents "très experts" ayant une connaissance importante de leur univers, des problèmes à traiter et des méthodes de résolutions (agents dits cognitifs). En effet, ces agents utilisent des processus de négociation/dialogue pour réaliser une éventuelle coopération ou la résolution d'un conflit. Ces dialogues, nombreux et complexes, nécessitent des capacités "cognitives" évoluées (langage, représentation de l'univers, mémoire) [Ferber, 1995].

Cette approche pose des inconvénients majeurs lorsqu'on souhaite l'appliquer à des agents situés (entité agissante dans un univers physique) :

- complexité des protocoles de communication et durée des négociations explosives en fonction du nombre (variable) des agents en interactions,
- temps important de réalisation des tâches (traitements sur des représentations symboliques de l'univers, planification des actions, dialogues, etc.)
- faibles performances pour des actions/réactions en temps réel (type navigation),

- agents inadaptés aux environnements dynamiques et/ou inconnus (agents lents, coûteux, fragiles et peu adaptatifs).

## L'intelligence collective

A la fin des années 80, une nouvelle voie d'exploration est apparue en IAD, répondant en partie aux problèmes soulevés précédemment. Il s'agit de l'étude et de l'utilisation des principes de fonctionnement des sociétés d'insectes. En effet, ce sont des systèmes composés d'agents très simples (*réactifs*) qui peuvent produire des constructions complexes et des solutions à des problèmes non triviaux (tris, parcours optimaux, répartition des tâches, etc. [Drogoul, 1993] [Bonabeau and Theraulaz, 1994]).

Les informaticiens ont repris les principes *d'auto-organisation* et *d'émergence* présents dans ces sociétés pour définir ce que l'on nomme l'intelligence collective (cf. [Bonabeau and Theraulaz, 1994]). La méthodologie principalement empirique qui l'accompagne - simulation, observation, perturbation - est toutefois renforcée par les acquis de l'éthologie et de la biologie.

Les avantages liés à l'utilisation d'une telle approche peuvent se résumer ainsi :

- fiabilité du système dans son ensemble (la perte d'un agent ne remet pas en cause le processus général),
- production d'une performance collective qualitativement supérieure à celle des unités,
- une plus grande flexibilité face aux aléas des environnements réels, tolérance aux pannes et utilisation des événements aléatoires.

Dans ce paradigme, la coopération est présente à travers *l'auto-organisation* des agents. Ce sont des comportements collectifs intelligents qui *émergent* des nombreuses interactions entre les agents et leur environnement.

Ces principes ont été repris pour la conceptions de robots autonomes réactifs. Il s'agit d'une part des systèmes composés de (nombreux) robots cellulaires [Beni and Wang, 1989] (concept de l'intelligence en essaim) et d'autre part de robots "à base de comportements simples" plus volumineux, non communicants et moins nombreux. Ces derniers ont montré leur aptitude à traiter des tâches de regroupement [Mataric, 1994a], de partage des ressources [Steels, 1994], de navigation en formation [Balch and Arkin, 1995], de transports "émergents" (BoxPushing) [Kube and Zhang, 1992] [Kube and Bonabeau, 2000] et de fourragement [Arkin, 1992a] [Mataric, 1995].

Cependant, l'approche collective ou réactive pose un certain nombre de problèmes lorsqu'elle est envisagée pour des applications réelles :

- difficultés pour anticiper la résolution d'un problème par une intelligence "émergente" (problèmes de formulation, de compréhension et de preuve de la résolution),
- nécessite un grand nombre d'agents (notion de masse critique), donc risques de conflits et de coût élevé,
- risques de comportements oscillatoires ou bloquants,
- pas de coopérations locales *intentionnelles*, c'est-à-dire de comportements volontairement coopératifs - décidés ou exigés par les individus -.

Il est clair que le paradigme collectif est délicat à manipuler et en particulier lorsqu'il s'agit de concevoir des robots mobiles autonomes chargés de réaliser des tâches complexes. La nécessité d'utiliser un grand nombre d'agents pose deux problèmes majeurs qui sont le coût financier et la chute des performances à partir d'un seuil de "surpopulation" [Drogoul, 1993] [Fontan and Mataric, 1996]. De plus, l'impossibilité d'assurer des coopérations intentionnelles limite les performances locales des agents (notion d'entre-aide, de partage direct d'information).

Pour répondre à ce problème, des solutions introduisant des *communications* ont été proposées. Par exemple, l'architecture ALLIANCE de L. Parker assure une répartition dynamique des tâches par échanges de messages simples [Parker, 1994]. Mais ce modèle ne permet pas de gérer d'autres activités de coopération.

Cette démarche s'apparente à la stratégie la plus employée pour faire coopérer des agents, qui consiste à définir des architectures hybrides communicantes. Ce sont des architectures introduisant une couche délibérative (et communicante) au dessus d'une couche réactive (cf. [Arkin, 1998]). Par conséquent, cette approche n'échappe pas aux difficultés et aux défauts, évoqués plus haut, des systèmes cognitifs.

## 1.2 Combiner coopération directe et indirecte...

Pour pallier ces difficultés, nous proposons une architecture définissant des agents à la fois réactifs et intentionnellement coopératifs. La résolution d'un problème distribué est alors obtenue par un ensemble de coopérations locales (directes) et une auto-organisation au niveau global (coopération indirecte).

Nous avons évoqué la nécessité de communiquer pour établir une coopération intentionnelle entre les agents. Mais cette capacité est souvent source de "complexité" et de

lenteur dans les systèmes. Ainsi, nous en avons conclu que la coopération intentionnelle devait être intégrée au niveau “réactif”.

Notre problème général peut alors s’exprimer par la question :

---

*Comment introduire une forme de coopération intentionnelle dans l’approche réactive ?*

---

Notre démarche est alors

- de définir la coopération en terme d’interactions simples et locales,
- d’exprimer ces interactions intentionnelles dans le “langage” de l’auto-organisation,
- d’en dégager une architecture réactive et coopérative.

Pour “abstraire” l’objectif, disons que nous cherchons à introduire les notions d’intentions et de coopérations au niveau réactif et non comme une sur-couche dominante. Ceci dans le but de développer une architecture coopérative conservant les atouts de l’approche réactive tout en limitant ses inconvénients.

Pour ne pas remettre en cause les capacités d’auto-organisation des agents, nous suivons les deux règles suivantes :

- limiter les actions coopératives au niveau *local* et s’assurer quelles sont *satisfaisantes* pour les agents en interactions,
- exprimer ces actions dans un modèle réactif.

La notion de satisfaction est essentielle dans notre approche, car elle détermine l’état de chaque agent et leurs besoins (insatisfactions). Notre travail s’appuie sur le développement de cette notion, qui nous permet d’exprimer l’état des agents, leurs besoins et l’évolution du système.

## Approches ascendante et descendante

Les travaux développés dans cette thèse sont menés suivant une démarche essentiellement ascendante. Dans un premier temps nous nous concentrons sur l’étude des mécanismes d’interactions pour définir des actions coopératives. De là, nous développons une architecture générique assurant un comportement adaptatif, coopératif et temps réel aux agents. Enfin nous spécialisons ce modèle pour traiter des problèmes concrets (cf. fig. 1.2).

Cependant, en termes de traitement de problèmes distribués, notre approche ne

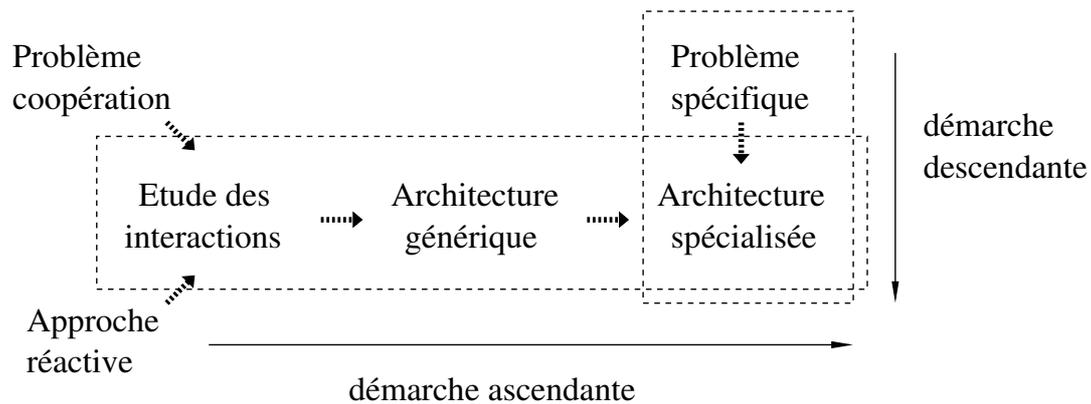


FIG. 1.2 – Démarches à la fois ascendante et descendante de notre étude

reste pas ascendante. En effet, la spécificité des problèmes abordés nous conduit à spécialiser l'architecture pour la rendre plus performante (cf. fig. 1.2). Cette phase de spécialisation nous mène à définir des règles comportementales locales provenant autant d'une réflexion descendante (les contraintes du problème et leurs conséquences au niveau local) qu'ascendante (emploi de notre architecture pour des problèmes spécifiques). Ceci est justifié par le fait que des règles locales peuvent, par effet de propagation, avoir des implications au niveau global (cf. par exemple les règles de navigation proposées pour des robots en milieu très contraints, chapitre 9).

### Contraintes technologiques :

Le travail présenté dans ce manuscrit propose des solutions informatiques à des problèmes distribués sur des environnements physiques. Il s'agit de développer un modèle d'agents capable de les traiter en univers simulé comme réel. Ainsi, nous nous sommes efforcé, tout au long de ce travail, de proposer des solutions qui tiennent compte des contraintes imposées par la robotique en développant des techniques multi-agents véritablement implémentables sur des robots réels.

## 1.3 Plan de la thèse

Le travail présenté dans ce manuscrit est composé de cinq parties :

**PARTIE 1 : Problématique**

Cette partie donne dans un premier temps quelques définitions et repères sur le domaine des systèmes multi-agents et précise la notion d'agent situé (chapitre 2). Dans le chapitre suivant, nous approfondissons le concept de coopération (dans les SMA situés) pour définir concrètement nos objectifs et l'approche menée dans la thèse (chapitre 3).

**PARTIE 2 : Modélisation des satisfactions**

Cette partie est consacrée d'une part à la modélisation de la satisfaction personnelle de l'agent (chapitre 4) et d'autre part à l'étude de ses interactions (chapitre 5). Cette étude conduit à la définition des notions de signaux de satisfaction interactive et de réactions altruistes.

**PARTIE 3 : Le modèle Satisfaction-Altruisme**

Nous développons une architecture d'agent fusionnant les concepts de satisfactions et d'altruisme définis dans les chapitres précédents (chapitre 6). Les algorithmes définissant ce modèle sont donnés en détails et analysés, enfin l'architecture est comparée à d'autres modèles d'agents existants (chapitre 7).

**PARTIE 4 : Application à la résolution de problèmes distribués et Simulations**

Le modèle proposé est tout d'abord expérimenté et évalué par simulation sur le problème classique des robots fourrageurs. Nous étudions plus en détails les paramètres de communication sur la résolution du problème (chapitre 8). Nous appliquons ensuite notre modèle à la résolution de conflits spatiaux en milieux contraints (chapitre 9). Enfin, nous étudions au chapitre 10 l'intégration de l'apprentissage dans cette architecture.

**PARTIE 5 : Communications et expérimentations en robotique mobile distribuée**

Nous présentons au chapitre 11 le développement d'un *protocole* de communication *dynamique* dédié aux systèmes multi-agents situés. Ce protocole, ainsi que l'architecture proposée, sont implémentés sur des mini-robots mobiles autonomes disposant seulement d'émetteurs/récepteurs de signaux infrarouges. Nous validons le modèle général par des expérimentations de résolution de conflits spatiaux (chapitre 12).

La thèse se termine par un bilan général, précisant les problèmes résolus, partiellement résolus ou encore sans réponses. Nous en déduisons un ensemble de propositions pouvant être développées pour améliorer ces travaux et ouvrir vers de nouvelles voies.

# Chapitre 2

## Les systèmes multi-agents situés

### 2.1 Les systèmes multi-agents

Les systèmes multi-agents recouvrent un grand nombre de domaines d'applications : résolution de problèmes distribués, systèmes d'interactions-communications, simulations de systèmes artificiels-vivants, agents pour le Web, robotique mobile, etc. Paradoxalement, cette explosion du domaine n'a pas été accompagnée d'un fort développement de définitions générales et formelles l'unifiant clairement. L'hétérogénéité et la complexité naturelle de ces systèmes en sont sans doute responsables (cf. [Ferber, 1995]).

Ainsi, nous introduisons le domaine des systèmes multi-agents en considérant les caractéristiques communes à toutes ces applications. Cette démarche, employée par G. Weiss dans [Weiss, 2000], permet de dégager deux caractéristiques fondamentales à tous les systèmes multi-agents :

- leur *distribution intrinsèque*, qu'elle soit spatiale, temporelle, sémantique (ontologies et langages) ou fonctionnelle (pour les capacités de perceptions, d'actions et de cognitions).
- leur *complexité naturelle*, au sens que les constituants de ces systèmes sont trop nombreux pour être contrôlés/traités par une seule entité.

Les systèmes multi-agents sont donc des systèmes distribués - à travers les *agents* qui les composent - et complexes - au sens du *nombre important des interactions et des paramètres* -. Les travaux qui leur sont consacrés se divisent en deux grandes problématiques : *comprendre* le fonctionnement de tels systèmes et *construire* de tels systèmes. Ces deux objectifs sont naturellement complémentaires et synergiques.

La *compréhension/modélisation* des systèmes complexes est un domaine de recherche à part entière plus ancien que celui des SMA. Les premiers travaux peuvent être attribués au physiciens qui cherchent à modéliser depuis les années 60 les systèmes dynamiques et complexes (systèmes chaotiques cf. [Prigogine, 1993]). Ces études, basées sur l'utilisation d'objets mathématiques liés à la physique, se poursuivent mais sont encore difficilement exploitables pour l'étude de sociétés d'agents.

L'informatique a proposé des outils tels que les automates cellulaires pour étudier les systèmes dynamiques discrets. Mais là encore, le fossé qui sépare la théorie des applications concrètes reste immense (à noter toutefois les travaux de simulations de “robots” cellulaires de G. Beni et J. Wang [Beni and Wang, 1989]).

Dans les systèmes multi-agents réactifs, les notions de compréhension et de constitution se rencontrent. En effet, l'intelligence collective est née de *l'observation* des sociétés d'insectes car l'intelligence n'est pas dans les agents mais dans leurs nombreuses interactions.

En reprenant ce principe d'auto-organisation il est possible de résoudre des problèmes sans maîtriser au préalable le déroulement exact de la résolution. Seules l'observation et l'analyse peuvent expliquer en détails l'évolution du système vers la solution.

Cette deuxième voie, que nous venons d'évoquer et que nous développons plus loin, est une démarche expérimentale qui conduit naturellement à l'approche purement constructiviste. De nombreux systèmes multi-agents ont concrètement été développés dans des domaines d'applications aussi divers que la simulation, les outils pour le commerce électronique, les réseaux de télécommunications, l'optimisation de processus industriels, la robotique. L'ensemble de ces réalisations, souvent accompagnées de modèles formels, a largement participé à l'évolution du monde multi-agents, complétant et justifiant les travaux théoriques.

### 2.1.1 Les différents types de systèmes multi-agents

Les systèmes multi-agents peuvent être classés suivant les grandes caractéristiques duales qui définissent les agents et leur organisation.

Ces systèmes peuvent être **réels** (vivants, robotisés) ou **simulés**. Nous aurons l'occasion de préciser au cours de ce manuscrit que cette distinction est fondamentale, en particulier lorsqu'il s'agit de passer de modèles théoriques ou simulés à la conception. L'opération inverse est tout aussi délicate, puisque la simulation n'est qu'un modèle de la

réalité, et qu'elle est sujette aux biais de programmation.

On distingue classiquement les systèmes **réactifs** et **cognitifs**. Les premiers sont composés de nombreux agents très simples (leur comportement est basé sur des réactions à des stimuli) qui n'ont pas de mémoire individuelle mais qui par leurs seules interactions développent une intelligence collective.

Les seconds sont des systèmes composés d'un faible nombre d'agents mais aux capacités cognitives importantes. Leur aptitudes délibératives s'appuient généralement sur des représentations symboliques de l'univers, des raisonnements à l'aide de systèmes formels et des communications de haut niveau. Leur vitesse d'exécution, à l'opposé des agents réactifs, est souvent lente mais permet d'exhiber des solutions théoriques optimales. Leur complexité intrinsèque en fait aussi leur fragilité et leur coût très élevé.

Depuis une dizaine d'années, ces deux écoles se sont rapprochées pour tenter de fusionner au travers des architectures dites **hybrides**. Ainsi, de nombreux travaux proposent des agents présentant des capacités réactives qui sont subordonnées à des couches cognitives. D'autre part, des techniques d'apprentissages (par renforcement, réseau de neurones) permettent d'augmenter les capacités limitées des agents réactifs purs.

Le développement de solutions collectives a permis de montrer que certains systèmes pouvaient accomplir des tâches sans aucune communication directe entre les agents, il est donc nécessaire de préciser les moyens de communications d'un système. On distingue les systèmes dits **communicants** (il existe un moyen de communication *direct* entre les agents), les systèmes **purements communicants** (le système est composé d'agents "abstraites" ne pouvant interagir que par des communications directes - réseaux, DCSP -) et les systèmes **à communications indirectes** (c'est l'environnement des agents qui sert de médium de communication, exemples : phéromones déposés par les insectes, modification du terrain par une construction). Les communications dans les systèmes multi-agents sont complexes à mettre en oeuvre (proportionnellement au nombre d'agents et à leur dynamique) mais néanmoins primordiales pour leur fonctionnement (elles sont au coeur du système d'interaction). Ce sujet est largement abordé dans la thèse, en particulier dans le cadre de la conception d'un système de robots mobiles autonomes communicant par signaux.

Les agents artificiels sont conçus dans le but d'effectuer des tâches et de résoudre des problèmes. Leur comportement doit être défini suivant l'environnement dans lequel ils évoluent et le type de problème qu'ils traitent. On peut toutefois, en ne considérant

seulement les agents et leur type d'interaction distinguer les systèmes **coopératifs** des systèmes **compétitifs**. Les premiers sont caractérisés par un objectif commun à tous les agents ou par des objectifs individuels compatibles. Dans ce cas, les agents sont conçus pour coopérer, collaborer ou simplement se coordonner dans leurs actions (exemples : systèmes multi-robots coopératifs [Mataric, 1994a][Parker, 1994][Arkin, 1998]).

A l'opposé, les systèmes compétitifs mettent en jeu des agents ou groupes d'agents en compétition pour des ressources non partageables. Cependant, notons que la plupart des systèmes présentent les deux types d'interactions à la fois (exemples : système proie-prédateurs, robots footballeurs).

Les agents évoluant en coopération ou en compétition au sein d'un même système peuvent être identiques (capacités, compétences), on parlera alors de système **homogène** d'agents. C'est souvent le cas des systèmes purement réactifs (la *redondance* des compétences assure une grande robustesse au système). Les systèmes présentant des agents aux caractéristiques différentes sont dits **hétérogènes**. Dans ce cas, les agents sont *spécialisés* (initialement) pour certaines tâches. La réalisation de certaines tâches peut même nécessiter la coopération d'agents aux compétences différentes et spécifiques.

La dernière caractéristique fondamentale définissant un système multi-agents concerne l'organisation des individus entre eux. La plus simple organisation, mais aussi la plus flexible, est l'absence totale de dépendances entre agents. Ces systèmes sont composés d'agents que nous qualifierons d'**individuels**.

Un premier pas vers une réelle organisation consiste d'abord à maintenir **groupés** les agents (cf. coopération chapitre suivant). La notion d'**équipe** est plus forte, elle consiste à définir des rôles et des dépendances entre les agents (cf. [Gutknecht, 2001]).

Enfin, il existe de nombreuses organisations définissant des **hiérarchies**, plus ou moins fortes, entre les agents.

Toutes les combinaisons possibles de ces caractéristiques définissent autant de types de SMAs (cf. [Ferber, 1995] pour une classification détaillée et exhaustive). Dans le cadre de cette thèse nous n'avons abordé que certaines "classes" de SMAs, dont les caractéristiques sont les suivantes :

- ce sont des systèmes multi-agents *simulés* (partie 4) *et réels* (multi-robots mobiles autonomes) (partie 5),
- nous avons développé une architecture d'agents *individuels*, à base *réactive*, renforcée de notions *cognitives* introduites au niveau réactif (parties 1 et 2),

- les systèmes étudiés/réalisés pour l'évaluation de cette architecture sont successivement *homogènes* (chapitres 8, 9, 10) et *hétérogènes* (chapitres 11),
- les agents *coopèrent* pour la réalisation de tâches individuelles (non nécessairement compatibles) et pour un objectif global commun (partie 3),
- les agents peuvent *communiquer directement* entre eux (*localement*), par *diffusion de simples signaux* (partie 4).

L'étude de tels systèmes est motivé par plusieurs objectifs : proposer des solutions SMA pour la conception de robots mobiles autonomes coopératifs et utiliser les principes collectifs du vivant pour résoudre des problèmes distribués et traiter les conflits entre agents.

### 2.1.2 Objectifs et problèmes du domaine

Nous avons donné un aperçu du domaine des SMA et précisé quels types de systèmes étaient concernés par notre étude. Nous pouvons maintenant spécifier plus en détails les problèmes liés à la *réalisation* et *l'utilisation* des systèmes multi-agents en général. Les (sous) problèmes à résoudre en SMA sont (adapté de G. Weiss [Weiss, 2000]) :

- comment décomposer/répartir les tâches entre plusieurs agents,
- comment définir les protocoles et les langages de communication (indissociables des SMA et de la résolution des problèmes qu'ils traitent),
- comment rendre les agents capables de représenter et raisonner sur les actions et les connaissances des autres agents,
- comment faire agir un agent de façon cohérente et le rendre capable d'évaluer ses interactions,
- comment les agents peuvent détecter et résoudre les conflits,
- comment concevoir concrètement de tels agents, avec quelles méthodologies,
- comment éviter les situations oscillatoires ou chaotiques,
- comment organiser ou faire s'organiser les agents (structures et rôles),
- comment décrire formellement ces systèmes et les interactions entre agents,
- comment obtenir des comportements menant à la résolution de problèmes et/ou à l'apprentissage des agents ?

Ces (sous) problèmes, nombreux et complexes, existent dans pratiquement tous les types de systèmes multi-agents. Les problèmes abordés dans la thèse concernent plus spécifiquement la définition des comportements et des *interactions d'agents situés réactifs*, *l'évaluation par les agents de leurs actions*, la *coopération* et l'auto-organisation des agents, *la résolution*

*collective de problèmes distribués, le traitement des conflits spatiaux dans les systèmes multi-robots et la définition d'un protocole de communications dédié à ces systèmes.*

## 2.2 Des agents situés

"Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed."<sup>1</sup>

[Maes, 1995]

Le travail mené dans cette thèse concerne essentiellement des systèmes composés d'agents **autonomes** et **situés** dans un environnement. Leur autonomie est à la fois physique (énergie, capteurs, effecteurs) et décisionnelle - l'agent décide seul - (l'homme n'intervient pas). Ce sont des entités qui existent au sein d'un environnement physique réel ou simulé, on parlera d'**agents situés**.

**Définition** (*Agent situé*) [Ferber, 1995] : On appelle agent *purement situé* une entité physique (ou éventuellement informatique si on la simule) qui :

1. se trouve situé dans un environnement,
2. est mue par une fonction de survie,
3. possède des ressources propres, sous la forme d'énergie et d'outils,
4. est capable de percevoir localement son environnement,
5. ne possède pratiquement aucune représentation de son environnement,
6. possède des compétences,
7. peut éventuellement se reproduire,
8. a un comportement tendant à satisfaire sa fonction de survie, en tenant compte des ressources, des perceptions et des compétences dont elle dispose.

Cette définition englobe des agents très divers : les êtres vivants (insectes, animaux, hommes), les robots autonomes, les entités des mondes réels ou artificiels simulés. Notre étude se concentre sur le développement d'agents situés artificiels - de type robots mobiles - évoluant en milieu naturel (réel ou simulé). Nous devons préciser quelles sont les

---

<sup>1</sup>Les agents autonomes sont des systèmes calculant qui évoluent dans des environnements dynamiques et complexes, ils perçoivent et agissent de façon autonome dans cet environnement, et se faisant, réalisent un ensemble de buts et de tâches pour lesquels ils ont été conçus.

motivations (points 2 et 8 de la définition) des agents situés développés/étudiés dans la thèse :

*Les agents tendent à réaliser les tâches qui leur ont été assignées tout en assurant leur "survie" dans l'environnement. De plus, ces agents sont coopératifs et participent à un même objectif global.*

### **l'agent situé ... un animat**

Les chercheurs en intelligence artificielle et en robotique s'intéressent depuis longtemps aux systèmes vivants (c'est d'ailleurs une démarche naturelle dans les sciences de l'ingénieur).

La conception d'agents réels ou artificiels a particulièrement progressé au contact de sciences telles que l'éthologie et la biologie. Il s'en dégage même un domaine à part entière depuis dix ans, la conception *d'animats* (cf. proceedings des conférences *Simulation of adaptive behavior : From Animals to Animats* 1991, 92, 94, 96, 98 et 2000). On peut classer ces travaux en deux catégories d'objectifs :

- concevoir une entité artificielle (ou animat) reproduisant le fonctionnement et les performances d'un animal,
- définir l'organisation (interactions, coopération, compétition) - des animats à partir des comportements observés chez les insectes et les animaux.

Tout au long de la thèse, nous ferons régulièrement référence au monde du vivant. C'est en effet une source d'inspiration et de comparaison très riche. Notons toutefois que notre objectif n'est pas d'analyser, de modéliser, ni d'implémenter les techniques de communications ou d'organisations des animaux et des insectes. Essentiellement, parce que ces techniques font l'objet de nombreux points d'incertitudes et de controverses, et qu'ils ne sont pas toujours adaptés aux contraintes technologiques et de coûts de la robotique. Notre ambition est seulement de reprendre quelques grands principes, souvent déjà étudié en IAD, et de s'en inspirer pour la conception d'agents organisés.

## **2.3 L'approche réactive**

L'approche réactive débute en informatique (IA) dans les années 80 avec les travaux de M. Minsky (cf. "The Society of Mind" [Minsky, 1986]). Il propose alors un modèle

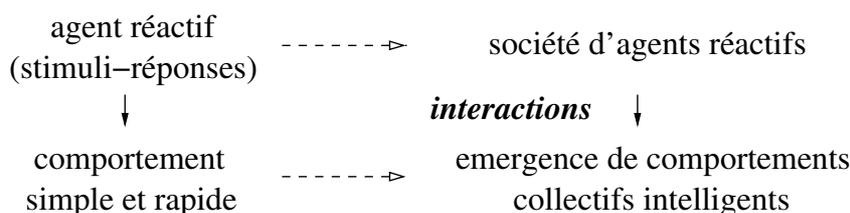


FIG. 2.1 – le comportement intelligent d’une société d’agents réactifs résulte de l’interaction de nombreux agents simples

du cerveau humain basé sur l’activité simultanée et voisine d’un nombre important de processus élémentaires, qu’il nomme agents.

Mais cette approche se développe réellement vers la fin des années 80, sous l’influence d’informaticiens et d’éthologues qui se penchent sur le fonctionnement des sociétés d’insectes (termites, fourmis, abeilles) ([Deneubourg et al., 1986] [Deneubourg and Goss, 1989], [Steels, 1989], [Drogoul, 1993]). Ils montrent que malgré la simplicité de chaque agent, la société parvient collectivement à résoudre des problèmes relativement complexes sans aucun traitement centralisé (tris, recherche de parcours optimaux, constructions d’objets tri-dimensionnels, répartition dynamique des tâches, localisation de lieux, etc. ), cf. principe fig. 2.1.

En terme de système multi-agents, on peut dire que c’est l’aboutissement de la pensée distribuée : le système est composé d’agents autonomes interagissant *localement*, qui n’ont pas de mémoire individuelle mais dont le système est lui même une mémoire collective dynamique et qui constitue, à certains états, le résultat d’un problème.

### Architecture réactive

Parallèlement aux travaux menés en informatique, la robotique voit se développer vers le milieu des années 80 des robots mobiles autonomes basés sur des architectures très simples. Il s’agit en réalité d’un retour en force de travaux plus anciens appartenant à la cybernétique ([Wiener, 1948]).

Historiquement, le premier “pseudo-robot” *mobile autonome* a été conçu sur le principe d’une architecture purement réactive, couplant directement perception et action. Cette réalisation est due à Walter Grey en 1953. Son “tortoise” robot était alors capable de se mouvoir de manière autonome, d’aller recharger ses batteries, de détecter une source lumineuse et de s’en approcher ou de s’en éloigner. Pour cela, le robot n’employait que deux

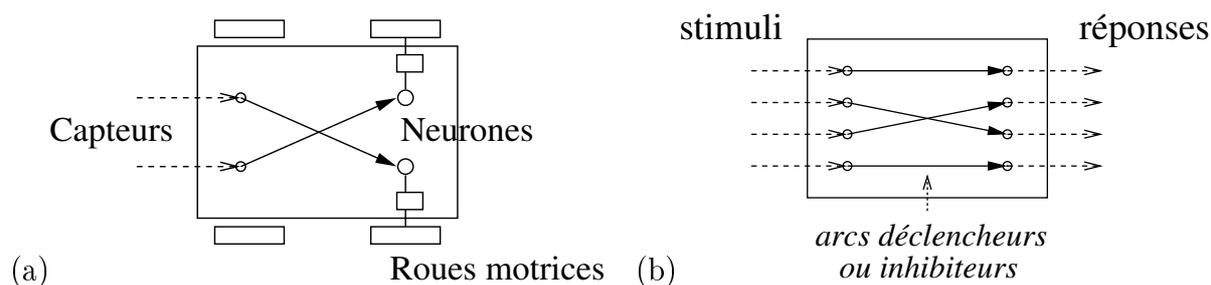


FIG. 2.2 – (a) exemple du véhicule de Braitenberg (b) principe d'une architecture stimuli-réponses

capteurs (lumière et contact), deux moteurs et deux tubes sous vide en guise de “cellules nerveuses” (cf. détails dans [Arkin, 1998]). Pourtant, malgré ces résultats encourageants, cette approche fut abandonnée durant 30 ans! Certainement parce que les travaux de la cybernétique ont dû faire place au développement de l'IA classique, privilégiant les méthodes délibératives basées sur des représentations symboliques de l'environnement.

En 1984 Valentino Braitenberg reprit le principe de W. Grey pour construire des robots mobiles autonomes capables de se diriger vers une source lumineuse à partir de deux capteurs et deux neurones (chacun activant un moteur proportionnellement à la force de l'information) [Braitenberg, 1984] (cf. fig. 2.2.a). Ces travaux furent étendus dans les années qui suivirent par R.A. Brooks [Brooks, 1986][Brooks et al., 1990][Brooks, 1990], R.C. Arkin [Arkin, 1987] et les chercheurs Hogg, Martin et M. Resnick [Hogg et al., 1991].

M. Mataric fut la première à réaliser des expériences “collectives” mettant en jeu plusieurs robots mobiles réactifs (basés sur les travaux de R.A. Brooks et R.C. Arkin) [Mataric, 1994a]. Les principes d'auto-organisation ont été appliqués et validés par l'accomplissement de certaines tâches telles que le regroupement, la poursuite, le maintien en formation et le fourragement. Nous détaillerons dans certaines sections des chapitres suivants les différentes tâches effectuées par ce type de systèmes.

Le comportement adaptatif et temps réel de ces robots mobiles en environnement naturel est du en grande partie au couplage direct réalisé entre les perceptions et les effecteurs (cf. fig. 2.2.b). C'est la notion d'*embodiment* introduite par R. Brooks en 1991 dans [Brooks, 1991] et initialement proposée en IA par Hubert Dreyfus [Dreyfus, 1979].

Il s'agit en réalité d'une démarche fondamentale, qui consiste à définir le robot (ou l'agent situé) en premier lieu comme *un corps en interaction avec son environnement* et non comme une entité abstraite calculante.

Cette prise en compte de l'existence du corps de l'agent a plusieurs conséquences :

- les agents n'ont pas à faire à des descriptions abstraites du système<sup>2</sup> mais avec les contraintes physiques et temporelles de la *réalité* ;
- ils ont des corps qui les mettent en rapport direct avec le monde, leurs actions *participent à la dynamique* du système, et par conséquent ces actions ont un effet immédiat sur leurs *perceptions* ;
- la source de l'intelligence de ces agents n'est pas limitée à un système de calcul, mais provient également de leur *situation*, de leurs *perceptions* et de leur *couplage* physique avec le système ;
- la conséquence générale c'est que l'intelligence émerge des interactions entre les agents et l'environnement, mais aussi des interactions directes entre les agents. Il est ainsi difficile de dire quelle est la source d'une action au sein du système.

Ces quatre conséquences peuvent être vues comme une définition des agents situés.

### De l'agent aux agents ...

Le prochain chapitre présente un niveau de complexité supplémentaire à la notion d'agent, celui de la coopération entre plusieurs agents. Nous allons voir que cette notion, très générale dans les SMA, est très dépendante de la définition même des agents. Toutefois, notre objectif est de reconsidérer certaines actions coopératives en termes d'actions spatiales, afin de les rendre exécutables par des agents réactifs.

---

<sup>2</sup>Le système est pris ici dans le sens général de monde, qui comprend un environnement et des agents.

# Chapitre 3

## Le problème de la coopération

### 3.1 Problème général

Avant de pouvoir proposer un modèle d'agent et une méthode de résolution de problème, nous devons préciser le type de tâche que nous souhaitons voir résolue par les agents.

Nous ne cherchons pas à résoudre des tâches ou des problèmes particuliers, mais à traiter des classes de problèmes. Il s'agit de problèmes distribués nécessitant l'emploi de plusieurs agents situés dans un environnement physique réel ou simulé. Les agents ne sont pas en compétition mais doivent coopérer soit par nécessité (tâche impossible pour un seul agent), soit pour des raisons de plus grandes performances. Ce sont par exemple des tâches de transports, de recherche d'objets, de constructions ou de navigation.

Hypothèses sur les agents :

1. ils ne détiennent pas de carte de l'environnement,
2. ils agissent de façon entièrement autonome (énergie, perception et décision),
3. ils sont pré-codés pour réaliser certaines tâches individuelles et certains comportements.

Quelles-sont les conséquences de ces hypothèses sur le traitement des problèmes distribués abordés? En premier lieu, la résolution ne passe pas par une planification globale des actions des agents. Mais par la *décomposition* du problème en *sous-tâches* pouvant être accomplies par un ou plusieurs agents.

Ainsi, les agents reçoivent des *compétences* leur permettant de participer à certaines de ces sous-tâches. Le dernier volet de la résolution ne peut être maîtrisé avant l'exécution du

système, il s'agit de *l'auto-organisation* des agents pour parvenir à l'achèvement de leurs sous-tâches et de l'objectif global.

Notre problème général peut maintenant être formulé par l'énoncé suivant :

---

Soit un système homogène ou hétérogène d'agents situés, évoluant dans un environnement dynamique qui leur est inconnu, *comment*

- *assurer la réalisation* des tâches/actions de chaque agent ?
  - *faire coopérer* les agents pour optimiser la qualité et le coût des tâches, mais aussi *éviter les conflits* ?
  - *faire s'adapter le système en temps réel aux variations* de l'environnement ?
- 

Pour développer un mécanisme de coopération, nous devons dans un premier temps définir cette notion et examiner les techniques existantes

## 3.2 Techniques de coopération

### 3.2.1 Définir la coopération

#### Le point de vue de l'observateur...

Les formes de coopérations possibles sont nombreuses et dépendantes du type de système multi-agents considéré. Ainsi, il n'existe pas de définition générale, mais plutôt des définitions partielles ou ad hoc aux diverses approches de l'IAD [Bouron, 1992].

Par contre, il est possible de poser des critères de performance permettant de mesurer les effets de la coopération sur un système (par comparaison avec des agents non coopératifs) [Bouron, 1992]. Ce sont des critères qualitatifs ou quantitatifs concernant la résolution d'un problème général (par exemple la quantité de minerai ramené par des robots fourrageurs sur une période de temps).

Ce type d'analyses a permis de dégager quatre catégories d'avantages liés aux comportements coopératifs ([Ferber, 1995]) :

1. Accomplir des tâches impossibles à réaliser seul,
2. Améliorer la productivité de chacun des agents,
3. Diminuer le temps de réalisation d'une tâche, où augmenter le nombre de tâches réalisées dans un délai imparti,
4. Améliorer l'utilisation des ressources : collaboration dans le but d'obtenir un des avantages 2 ou 3.

### Le point de vue de la biologie...

**Définition** (Coopération entre membres d'une même espèce) : l'interaction de deux ou plusieurs membres d'une même espèce qui produit un effet ou des conditions bénéfiques aux participants ou aux autres membres de l'espèce.

Nous retiendrons de cette définition que la coopération provient de *l'interaction* des agents et qu'elle *produit un effet bénéfique* sur les agents (cf. développements dans [McFarland, 1987]).

### Le point de vue de l'agent intentionnel...

La coopération peut être définie comme le comportement intentionnel d'un ou de plusieurs agents *s'engageant* dans des actions communes. Ce processus se définit en trois phases :

1. identifier un but commun,
2. adopté un but commun,
3. effectuer une action commune.

Ce type d'organisation, pour parvenir à l'accomplissement d'une tâche commune, est généralement réservé aux agents cognitifs, car elle requiert des communications de haut niveau et des engagements durables [Ferber, 1995].

Nous examinons maintenant les techniques de coopération permettant d'obtenir ces formes de coopérations.

### 3.2.2 Les méthodes de coopération

Les méthodes de coopération généralement employées sont résumées par le tableau 3.1. Nous avons précisé pour chaque méthode le type de système pouvant l'utiliser et les techniques employées.

**Regroupement et multiplication** La simple agrégation des agents est une forme de coopération (elle assure aux animaux une meilleure sécurité, plus de chaleur et le partage de la nourriture). Le regroupement simplifie la navigation de nombreux agents (volée d'oiseaux ou bancs de poissons). Il suffit qu'un seul agent (ou un petit groupe) décide de la direction à prendre et tous les autres n'ont qu'à le suivre par un comportement local simpliste (cf. [Reynolds, 1987]). La multiplication des agents, autrement dit la *redondance*

méthode	système	technique
regroupement	réactifs / vivants	réflexe / perception
spécialisation	réactif / délibératif	par l'environnement
arbitrage	délibératif / réactif	pré-codé
négociation	délibératif	communication
répartition des tâches, des info. et des ressources	délibératif réactif	comm. directe comm. indirecte
coordination d'actions	délibératif réactif	comm. directe / pré-codé perception / marques

TAB. 3.1 – Les méthodes de coopération

des compétences, assure une grande fiabilité (ou robustesse) au groupe. En contrepartie, tout regroupement peut dans certain cas critiques créer des situations d'encombrement et de conflits.

**Spécialisation** C'est un processus qui conduit un agent à progressivement se spécialiser dans certaines de ses tâches. Il s'agit d'une adaptation dynamique de l'agent aux conditions et aux besoins de son environnement (généralement dans les systèmes réactifs cf. [Deneubourg et al., 1991][Drogoul, 1993]).

**Arbitrage et négociation** Ce sont deux moyens de gérer les conflits entre agents. L'arbitrage établit des règles sur le comportement des agents qui ont pour conséquences au niveau global de limiter les conflits. La négociation permet aux agents cognitifs de résoudre des conflits. Ceux-ci s'engagent dans une véritable "discussion", nécessitant un système de communication de haut niveau, afin de parvenir à une solution ou à un compromis.

**Répartition des tâches, des informations et des ressources** Il s'agit des processus collaboratifs permettant aux agents de répartir les tâches, les informations et les ressources dans le but de réaliser un objectif commun. Cette répartition peut se faire au sein de systèmes délibératifs par des mécanismes d'offre et de demande (cf. détails dans [Ferber, 1995] et [Weiss, 2000]). Dans les systèmes réactifs, cette répartition se fait par le biais de l'environnement, et conduit à la *spécialisation* des agents et à leur répartition géographique.

**La coordination d'actions** “.. dans le cadre de la coopération, elle peut être définie comme l'articulation des actions individuelles accomplies par chacun des agents de manière à ce que l'ensemble aboutisse à un tout cohérent et performant” J. Ferber [Ferber, 1995].

Ce sont toutes les tâches ou actions qui ne sont pas directement productives mais qui assurent l'accomplissement de celles qui le sont. On distingue quatre formes de coordination d'actions :

- coordination par **synchronisation** (recherche d'un enchaînement cohérent d'actions non parallélisables, très présent dans les systèmes automatiques industriels et les systèmes d'exploitation répartis),
- coordination par **planification** (nécessite soit un agent coordinateur qui planifie les actions de chaque agent, soit l'échange et la construction de plan partiels dans le but de converger vers un plan global [Durfee and Lesser, 1991], [Durfee, 2000]),
- **coordination réactive** (aucune planification, ce sont des agents réactifs qui s'auto-organisent au travers de leur interactions avec l'environnement),
- coordination par **réglementation** (règles de comportements visant à éviter les conflits).

Notre objectif de développement d'agents réactifs coopératifs nous conduit à présenter plus en détails les techniques de coordination réactive.

### Coordination réactive

Les méthodes de coordination réactives se résument presque toutes à l'utilisation des techniques suivantes :

1. Utilisation de champs de potentiels ou plus généralement de champs de vecteurs pour la détermination du déplacement des agents mobiles.
2. Utilisation de marques pour coordonner l'action de plusieurs agents, ces marques permettent d'exploiter l'environnement comme un système de communication souple, robuste et simple (cf. présentation détaillée section 8.2.1.2).
3. Utilisation de l'éco-résolution [Ferber, 1989] : c'est une méthode de résolution par coordination qui consiste à *reformuler* un problème en un ensemble d'agents en interaction - les éco-agents - qui tentent de satisfaire individuellement leur propre but. Chaque éco-agent tente de réaliser son but, et s'il perçoit un gêneur il l'agresse. Un agent agressé tente de fuir en tenant compte des contraintes de son agresseur. Ce schéma se reproduit et fait évoluer le système jusqu'à ce qu'il aboutisse à un état stable (satisfaction de tous les agents), la solution du problème.

Il existe tout un ensemble d'applications de ces techniques. les plus remarquables sont la gestion du trafic aérien (par champ de potentiels et forces de glissement [Zeghal, 1994]),

l'organisation de robots-fourmis (par déposé de marques [Drogoul, 1993]), la navigation de groupes de robots ([Mataric, 1994a], [Arkin, 1998]) et le traitement de problèmes par éco-résolution ([Bura et al., 1991], [Drogoul, 1993]).

Avantages et défauts de ces techniques :

- L'emploi des champs de potentiels (ou de vecteurs) permet de traiter en temps réel le problème de la navigation d'un agent ou d'un groupe d'agents, sans utiliser de communications, mais seulement des perceptions locales. En contrepartie, cette technique se limite à la gestion des problèmes de navigation car il est difficile de représenter d'autres notions que celles d'obstacle ou d'attracteur à partir de perceptions locales de l'environnement.
- Le dépôt de marques permet de créer une véritable dynamique entre les agents et l'environnement (comportements collectifs intelligents émergents). Les agents peuvent s'échanger des informations et construire des chemins sans aucune communication directe. Par contre, cela nécessite un grand nombre d'agents (une masse critique). Son emploi reste donc difficilement envisageable pour des robots réels, de part le coût en robots et le coût pour déposer et reprendre de nombreuses marques (cf. développement chapitre 8).
- L'éco-résolution : c'est une technique de résolution complètement distribuée (pas d'explosion combinatoire) qui résiste et exploite le bruit (les perturbations). Par contre, son application nécessite de reformuler le problème en éco-agents. Ces agents ne peuvent être que dans des relations de buts personnels / agressions.

### 3.2.3 Bilan

Cette description des techniques de coopération cognitives et réactives nous conduit à plusieurs conclusions :

- les actions de coopération intentionnelle sont bénéfiques pour le système et/ou les agents mais elles nécessitent un moyen de communication *direct*,
- les systèmes collectifs s'auto-organisent pour coopérer à des tâches globales, mais ils sont incapables de coopérer directement,
- la coordination réactive permet de traiter des problèmes locaux (navigation) et d'agir au niveau global par propagation de contraintes.

Notre démarche consiste à exprimer les coopérations intentionnelles dans un modèle de coordination réactive. Le but étant de définir des agents intégrant les deux approches de coopération.

## 3.3 Principe de notre approche

### 3.3.1 Actions coopératives

Un agent situé (réactif) a pour but d'accomplir les tâches individuelles pour lesquelles il a été programmé. Supposons qu'il puisse accomplir des *interactions* indépendantes de ses tâches personnelles. Il pourrait alors agir "intelligemment" pour *améliorer* la réalisation des tâches de ses voisins ou simplement sa situation courante.

Nous avons nommé - actions coopératives - ces interactions particulières. Nous les définissons ainsi :

**Définition** (*action coopérative*) : On appellera action coopérative d'un agent A envers un autre B toute interaction qui améliore qualitativement ou quantitativement la tâche de B ou leur tâche commune.

Pour déterminer concrètement ces actions coopératives, nous avons tout d'abord recherché ce qui définissait l'état courant d'un agent. Puis nous avons dégagé les comportements coopératifs pouvant améliorer ces états.

Généralement, l'*état* d'un agent situé peut être déterminé par les trois critères suivants :

- l'avancement de sa *tâche courante* : progression ou blocage/panne,
- l'état des *ressources* utilisées : abondantes, limitées ou absente,
- le type d'*interaction* avec ses voisins : neutre, aide ou conflit.

Nous avons déduit de la combinaison de ces critères un ensemble de situations génériques nécessitant des *actions coopératives*. Nous avons ainsi posé cinq couples {situation, action coopérative} nous apparaissant comme fondamentaux, ils sont donnés dans le tableau 3.2.

Commentaire du tableau :

*Les deux premières lignes* expriment des situations où l'agent a besoin d'autrui pour réaliser une tâche ou simplement rester "en vie". Si un ou plusieurs agents lui viennent en aide, il s'agit d'actions coopératives améliorant qualitativement la situation des agents (et quantitativement celle du système).

*La troisième ligne* exprime une coopération intentionnelle consistant à partager une ressource abondante ou des informations pertinentes. L'objectif étant l'amélioration quantitative des performances des agents.

*La quatrième ligne* représente au contraire le risque d'un conflit lorsqu'un agent détient une ressource limitée. Dans ce cas, il agit coopérativement s'il tente d'éviter un conflit avec d'autres agents.

situation de l'agent	action coopérative de l'agent	action coopérative d'autrui
tâche impossible sans aide		aider à la réalisation
agent en panne/bloqué		débloquer - dépanner
détenir/connaître une ressource abondante	partager / informer les autres	
détenir une ressource limitée	éviter le conflit	
être en conflit avec un voisin	résoudre le conflit	résoudre le conflit

TAB. 3.2 – Les cinq couples de référence situation - action coopérative.

Enfin, *la dernière ligne* correspond à la résolution directe des conflits par des actions. Notons que les actions coopératives peuvent provenir simultanément de tous les agents impliqués.

De ce tableau, nous pouvons dégager les moyens minimaux nécessaires aux agents pour réaliser des actions coopératives.

Les agents doivent être capables :

1. de **percevoir** une situation nécessitant une action coopérative (besoin d'aide d'un agent, conflit),
2. d'**informer** les autres agents (pour partager des ressources ou des informations),
3. d'agir pour **aider** ou pour **résoudre/éviter un conflit**.

Le premier point, celui de la *perception par un agent* d'une situation nécessitant une action coopérative, est relativement complexe à mettre en oeuvre. Généralement, les agents utilisent un *moyen de communication* pour s'informer des pannes, partager des informations (point 2) ou résoudre des situations conflictuelles (point 3).

Il s'agit généralement de communications directes, mais le partage d'informations sur les ressources peut se faire de façon indirecte.

Le problème certainement le plus complexe est de concevoir des agents capables de s'aider lorsque la situation le nécessite (point 3). Cette notion n'existe pas au sens *intentionnel* dans les systèmes collectifs où la réussite de la société importe plus que la vie de chacun de ses membres. Mais dans un système de robots collectifs, le coût de chaque robot n'est pas négligeable et cette notion d'aide et d'évitement des pannes est primordiale.

### 3.3.2 Coordination réactive et actions coopératives

Nous venons de définir les situations types nécessitant des comportements coopératifs. Nous avons aussi déterminé les capacités minimales pour détecter ces situations et agir en conséquence. Il faut maintenant “classer” toutes ces situations pour en déduire une stratégie comportementale coopérative.

#### Les pannes

Les raisons principales pouvant engendrer la panne d’un agent sont :

- impossibilité d’accéder à une borne/source d’énergie,
- un choc avec un autre agent mobile,
- un choc avec un obstacle,
- un élément interne défectueux.

Il est important de noter que les trois premières causes de pannes sont liées aux capacités de *navigation* de chaque agent (chocs avec l’environnement) et de *navigation collective* (chocs entre agents et nécessité de pouvoir atteindre une source d’énergie).

#### Les conflits

Les conflits entre agents sont liés à l’utilisation simultanée d’une même ressource, de type *objet, espace ou énergie*.

**Définition** (*conflit pour des ressources*) : On appelle situation de conflit pour des ressources, des agents ou des groupes d’agents qui ont besoin simultanément des mêmes ressources sans vouloir les partager.

Le conflit peut mener à une situation de *blocage*. Si les efforts mis en jeu par les agents sont égaux, il se produit un état d’équilibre bloquant. On peut se représenter une telle situation par deux personnes qui tirent de chaque côté d’une porte avec la même intensité.

Les agents situés, donc mobiles et autonomes, sont par nature susceptibles d’entrer en conflit pour l’accès à des zones de l’espace ou à des ressources en général. A nouveau, la navigation est à la source d’un problème potentiel pour les agents (le chapitre 9 est consacré au traitement des conflits spatiaux en milieu très contraint et à leur résolution).

### 3.3.2.1 Naviguer avant tout...

Nous considérons la navigation individuelle ou collective des agents comme prépondérante dans le fonctionnement d'un SMA situé. Nous venons de voir que des points critiques tels que les pannes ou les conflits dépendaient des capacités des agents à *se déplacer* "intelligemment".

De plus, les tâches individuelles (transport, recherche d'objets, constructions) ou les actions coopératives (venir en aide, transport à plusieurs agents, évitement et résolution des conflits spatiaux) nécessitent des actions de déplacements.

Par conséquent nous allons exprimer les actions coopératives en terme de déplacements et placer le système de coopération des agents directement au niveau de la navigation.

L'objectif étant de définir une architecture traitant prioritairement

1. les conflits de trajectoires / d'espace,
2. les coopérations spatiales et les transmissions d'informations,
3. la réalisation des tâches individuelles spatiales, puis stationnaires.

## 3.4 Organisation des chapitres suivants

L'élaboration de l'architecture proposée s'est faite en considérant l'agent à des niveaux de complexité incrémentaux : -individuellement- puis avec -son voisinage- et enfin dans -le système- (approche que l'on retrouve dans la méthode *Cassiopee* [Collinot et al., 1995]).

Cette démarche s'explique par la nature même des agents recherchés, ils doivent

- accomplir des tâches personnelles,
- effectuer des actions coopératives locales (entre voisins),
- s'auto-organiser pour résoudre un problème général.

Nous présentons dans un premier temps une étude de "l'état de satisfaction" des agents, qui est fonction de la progression de leurs tâches individuelles, et dont nous tirons le concept de satisfaction personnelle.

Puis, pour définir des comportements coopératifs intentionnels, nous posons les notions de signaux de satisfaction interactive et d'altruisme.

Ainsi, nous proposons une architecture réactive intégrant des comportements coopératifs intentionnels, dont le système décisionnel est contrôlé par des "flux de satisfactions".

Le tableau ci-dessous résume l'organisation de ces travaux à travers les chapitres de la thèse.

Niveau	Étude	Chapitre(s)
Agent	Approfondir la notion d'état de satisfaction	4
Interaction	Définir la satisfaction interactive et proposer un mécanisme simple pour réaliser des actions coopératives	5
Système	Définir une architecture d'agents situés réactifs et intentionnellement coopératifs	6, 7



## Deuxième partie

### Modélisation des satisfactions



# Chapitre 4

## Modélisation de la satisfaction personnelle d'un agent situé

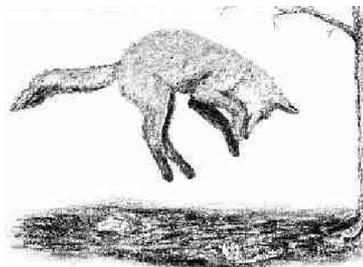
L'INTELLIGENCE ARTIFICIELLE est un domaine de recherche visant à faire à des machines ce qui, d'après les humains, requiert de l'intelligence. Il n'y a pas de frontière nette entre la psychologie et l'intelligence artificielle, car le cerveau est lui-même un type de machine.

M. Minsky

### 4.1 Introduction

Considérons un *petit animal* qui suit un comportement régi par le simple principe :

$$\textit{insatisfaction} \Rightarrow \textit{action} \Rightarrow \textit{satisfaction} \quad (4.1)$$



Lorsque l'animal est dans un état ou une situation insatisfaisante il agit de telle sorte que la situation s'inverse pour tendre vers un état de satisfaction. Cette expression du comportement est développée dans le modèle homéostatique ou théorie pulsionnelle de

C.Hull [Hull, 1943]. Le principe est donné par le schéma suivant :

$$\begin{aligned} \textit{Equilibre} &\Rightarrow \textit{equilibre rompu} \Rightarrow \textit{besoin} \Rightarrow \textit{pulsion} \Rightarrow \textit{comportement} \\ &\Rightarrow \textit{satisfaction du besoin} \Rightarrow \textit{retablissement de l'équilibre} \end{aligned}$$

Dans le cadre d'agents vivants (animaux ou humains) les insatisfactions ou besoins sont dus à des *motivations* qui organisent et assurent la survie de l'individu. Les principales motivations pour un animal sont d'ordre physiologiques :

- éviter les prédateurs, la douleur,
- se nourrir,
- se reproduire.

Le dessin donné ci-dessus représente un renard en train de sauter sur sa proie. Celui-ci, ayant faim, a entrepris un ensemble d'actions qui le mène jusqu'à la capture et la consommation d'une proie.

De nombreux travaux ont été consacrés aux motivations et à leur implications sur le comportement (cf. théorie de A. Maslow pour le cas de l'homme [Maslow, 1943] et [McFarland, 1987] pour une introduction au cas des animaux).

Dans le cadre de notre étude, nous reprenons ce modèle pour définir le comportement d'entités artificielles tendant à être satisfaites dans l'accomplissement de leurs tâches.

### **les motivations de l'agent**

En général, un agent artificiel n'est pas conçu pour reproduire le comportement d'un animal, mais effectuer les tâches attendues par le concepteur. Ces tâches définissent les motivations de l'agent. Certaines s'apparentent à celles des animaux (rester "vivant" : gestion de l'énergie, éviter les chocs, éviter les conflits avec d'autres agents), d'autres sont simplement la nécessité de réaliser des tâches lorsque la perception d'une situation l'exige. Dans ce cas nous parlerons de *situation de tâche* (ou de perception des stimuli déclencheurs d'une tâche).

Dans les systèmes multi-agents, chaque agent est soumis à des motivations d'ordres internes et externes. J. Ferber propose une décomposition des motivations en 3 grandes composantes : individuelles, relationnelles et environnementales [Ferber, 1995] (cf. représentation figure 4.1).

Dans le cadre de nos hypothèses, les motivations personnelles de l'agent sont le besoin (l'obligation) d'accomplir les tâches qu'il détecte comme étant réalisables et de persister

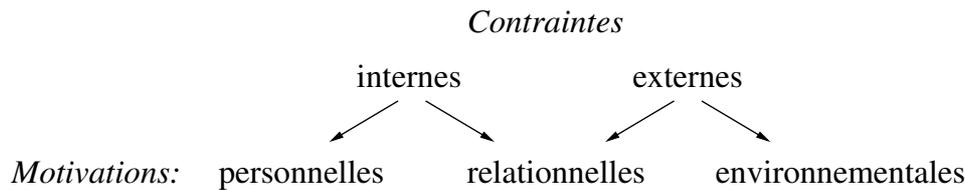


FIG. 4.1 – Décomposition des motivations d'un agent évoluant dans un SMA

dans leur accomplissement.

Les motivations ou contraintes relationnelles sont définies par les engagements qu'un agent peut passer envers d'autres et qui le poussent à accomplir des actions/tâches (notion de contrat). Cette notion d'engagement est essentiellement présente dans les systèmes multi-agents cognitifs.

Les motivations ou contraintes environnementales sont les perceptions du monde extérieur qui déclenchent ou poussent un agent à réagir (concepts de stimuli et réflexes). Par exemple la perception d'un prédateur pousse un animal à immédiatement s'éloigner.

Le principe des motivations personnelles et environnementales a été repris ces dix dernières années pour la conception de différentes architectures d'agents réactifs. Les plus importants travaux sont ceux de P. Maes (Action-Sélection) [Maes, 1991], A. Drogoul (EMF) [Drogoul, 1993], R.C. Arkin (Motor Schemas) [Arkin, 1987][Arkin, 1989][Arkin, 1998], M.J. Mataric [Mataric, 1994a] et L. Parker (ALLIANCE) [Parker, 1994].

Notons que R.C. Arkin a introduit en 1992 la notion de contrôle homéostatique dans son architecture [Arkin, 1992b]. Ainsi les comportements d'un robot sont influencés par la perception de déséquilibres ou de besoins internes. Plus récemment, il a défini avec A. Stoytchev une architecture mono-agent hybride utilisant un ensemble de variables de motivations (frustration, besoin d'énergie, curiosité et colère) [Stoytchev and Arkin, 2001]. L'évaluation de ces variables se fait par *perception de l'échec/réussite des tâches* et varient linéairement avec le temps (incrémentalement ou décrétement).

### des satisfactions aux motivations...

Si les modèles de motivations pour la sélection de tâche et la spécialisation sont très développés, les modèles de persistance liée à la *progression de la tâche en cours* sont souvent limités à de simples mesures du temps écoulé (notion de frustration) ou de la perception d'un but accompli ou non.

Nous présentons ci-après une étude approfondissant la notion de satisfaction liée à la

progression des tâches personnelles. Le chapitre suivant est consacré à l'étude des satisfactions relationnelles et nous conduit à introduire la notion d'engagement dans les systèmes réactifs. Ainsi, ces modélisations de satisfactions nous permettent d'intégrer véritablement les trois types de motivations au coeur d'une architecture d'agent réactif.

## 4.2 Mesures de la progression des tâches

### 4.2.1 Tâches personnelles

La plupart des travaux, théoriques ou pratiques, menés sur la conception d'agents situés, cherchent à maximiser la qualité de réalisation des tâches et à minimiser leur coût en temps. Les solutions optimales en temps peuvent être obtenues par des approches théoriques mais elles ne sont pas toujours applicables. En effet, par nature, ces solutions ne prennent pas en compte la totalité des paramètres des situations réelles et de leur caractère dynamique [Mataric, 1994a], [Pirjanian, 1999]. Par exemple, le calcul d'un chemin optimal entre deux points de l'environnement peut se révéler à l'exécution inadapté à la nature du terrain, obsolète face à la réalité des obstacles, etc.

Aussi, une bonne solution à un problème n'est pas nécessairement la plus optimale en temps (ou en énergie) mais celle qui garantit un résultat. Si l'on considère des agents autonomes évoluant dans un environnement *dynamique* et *inconnu*, la notion de temps perd de son importance au profit des capacités de réalisation et d'adaptation des agents. Dans ce type de système, un agent risque plus souvent d'être confronté à une situation inconnue qu'à une situation idéale où l'exécution des opérations peut être considérée comme déterministe.

### 4.2.2 Notion d'exécution de tâche

Les agents situés définis pour le traitement de problèmes globaux ou de sous-problèmes ont à leur disposition un ensemble de compétences (ou fonctions) données par leur concepteur.

Généralement, chacune des fonctions  $f$  de l'agent est accompagnée d'une condition de déclenchement  $d_f$ . La fonction peut aussi détenir une condition de terminaison  $t_f$ . Nous écrivons l'exécution d'une fonction :

$$d_f \Rightarrow \{executer(f) \text{ tant que } \neg t_f\}$$

Certains agents, dits réactifs purs, agissent de manière totalement réflexe, suivant le

schéma stimulus/réponse. Dans ce cas la condition de terminaison n'existe pas, car l'action exécutée est généralement simple et de courte durée (on parle d'action élémentaire ou atomique). L'exécution d'une telle action s'écrit simplement  $d_f \Rightarrow \text{executer}(f)$ .

En dehors des agents réactifs purs, les tâches des agents sont souvent décomposées en fonctions ou primitives de tâches possédant des conditions de déclenchement, de terminaison et de contrôle. L'exécution d'une fonction  $f$ , qui doit mener l'agent à la réalisation d'un (sous) but, peut être contrôlée par une mesure de la distance au but (notée  $D(f)$ , qui donne à un instant donné cette distance). Notons *progres* la fonction qui détermine la progression ou non de l'agent vers son but entre deux instant consécutifs :

$$\text{progres}(x) = \begin{cases} \text{si } (x_t - x_{t-\Delta t}) < 0 & \text{alors vrai} \\ \text{sinon} & \text{faux} \end{cases}$$

Cette fonction peut servir à décider de l'abandon de la tâche en cours en cas de non progression (stagnation ou recul de la distance au but). Ainsi, nous écrivons :

$$d_f \Rightarrow \{\text{executer}(f) \text{ tant que } (\neg t_f \text{ et } \text{progres}(D(f)))\}$$

ou plus clairement :

$$d_f \Rightarrow \text{executer}(f, D) : \begin{cases} \text{si } t_f & \Rightarrow \text{accomplie} \\ \text{si } \neg \text{progres}(D(f)) & \Rightarrow \text{abandon} \\ \text{sinon} & \text{continuer} \end{cases}$$

Ce principe de base, souvent employé pour sa simplicité, cache des difficultés :

- le problème de la perception de la réussite de la fonction (évaluation de  $t_f$ ),
- l'abandon rapide en cas de non progression de la fonction.

Le premier problème est inhérent à tous les systèmes artificiels, il relève autant de la robotique que de l'informatique (perception, représentation et analyse d'une situation). C'est sur le second problème que se concentre notre travail. Nous définissons en section 4.3 un niveau de satisfaction qui intègre des informations dans le temps permettant à l'agent de reculer ou stagner temporairement sans abandonner nécessairement sa tâche en cours (remarque : démarche opposée à celle d'une heuristique tabou).

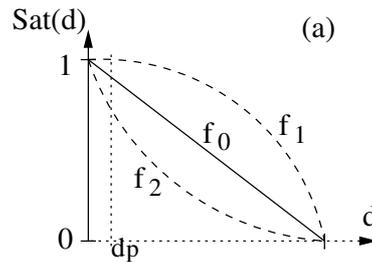


FIG. 4.2 – Courbes de satisfaction fonctions de la distance au but

### 4.2.3 Mesures classiques de la progression des tâches

Généralement, un agent évalue sa satisfaction (ou la récompense dans le cas d'un apprentissage) par la perception de la *réussite*, de l'*échec* ou de la *durée relative* de la tâche exécutée, c'est-à-dire *en fin de tâche*. Au contraire, notre objectif est de déterminer un niveau de satisfaction *pendant l'exécution* de la tâche.

Des évaluations en cours d'exécution existent, elles sont généralement définies par une satisfaction ou une motivation fonction de la distance au but (ex. [Horswill, 1995]) ou de la force d'un gradient [Arkin, 1998]. Mais ces méthodes ont l'inconvénient d'exprimer seulement une *proximité* au but, sans référence aux actions passées. La figure 4.2 donne des exemples de fonctions de satisfaction pour un agent devant atteindre un lieu particulier de l'espace. La satisfaction est très élevée quand l'agent est proche du but (distance  $d_p$ ) alors qu'il peut être bloqué ou obligé de "reculer" temporairement. Pour pallier ce problème, des mesures *d'impatience* ont été proposées (ex. [Parker, 1994]). Elles sont efficaces lorsque la durée est le critère principal de satisfaction, mais elles ne permettent pas d'évaluer qualitativement la tâche en cours.

### Évaluation des tâches en apprentissage

Les méthodes d'apprentissage par renforcement nécessitent des fonctions d'évaluation ou de récompense des comportements des agents. Dans les système non supervisés, les agents doivent déterminer par eux-même cette valeur de récompense. Généralement, cette évaluation est effectuée en fin de tâche par la perception locale de l'environnement de l'agent. Mais toutes les tâches ne se prête pas à ce type d'évaluation [Mataric, 1994b].

De ce constat, M.J. Mataric a proposé dans le cadre de l'apprentissage par renforcement de robots mobiles autonomes des mesures nommées *Progress Estimators* [Mataric, 1994a] [Mataric, 1994b]. Ce sont des fonctions associées à des comportements particuliers, qui fournissent une mesure en continue de la performance de l'agent. L'intérêt est de faire

décroître la sensibilité de l'apprentissage au bruit et de minimiser la probabilité de recevoir des récompenses aberrantes.

Concrètement, il s'agit d'une mesure de la progression (ou régression) de la tâche en cours d'exécution. Cette mesure est effectuée à chaque pas de temps et peut prendre deux formes :

$$R_1(t) = \begin{cases} m & \text{si progression but} \\ n & \text{sinon} \end{cases}$$

$$m > 0, n < 0$$

$$R_2(t) = \begin{cases} m & \text{si progression but} \\ f & \text{si éloignement but} \\ 0 & \text{sinon} \end{cases}$$

$$m > 0, f < 0$$

Ces deux types de mesures sont utilisés pour détecter les situations d'échecs et calculer le renforcement des tâches.

Dans le premier cas, une absence de progression de la tâche courante déclenche son arrêt :

$$\text{Si } R_i(t) < \text{SeuilNeg alors stopper tâche}$$

Il s'agit d'un simple test comparant la valeur courante  $R_i(t)$   $i \in 1, 2$  avec un seuil représentant l'échec de la tâche. Notons que ce type de test reste sensible à une erreur de mesure en provenance des capteurs. Nous proposons ci-après une évaluation de l'exécution des tâches résistant mieux à ce problème.

Le deuxième emploi des *progress estimators* est leur application au calcul de la valeur de renforcement de l'apprentissage :

$$A(b) = \sum_{t=1}^T R_i(t)$$

Le comportement  $b$  est évalué en sommant les valeurs obtenues sur la durée d'exécution. Ainsi, de faibles erreurs de mesures sont naturellement absorbées par ce calcul de moyenne. La nouvelle valeur  $A(b)$  renforce l'ancienne, elle représente la probabilité de déclencher l'action  $b$  lorsque les conditions de déclenchement sont vérifiées.

Dans notre étude, nous avons cherché dans un premier temps à évaluer un état de satisfaction de l'agent dépendant uniquement de l'exécution de sa tâche courante (cf. principe figure 4.3). Notre stratégie, comme dans les *progress estimators*, est de réaliser une mesure

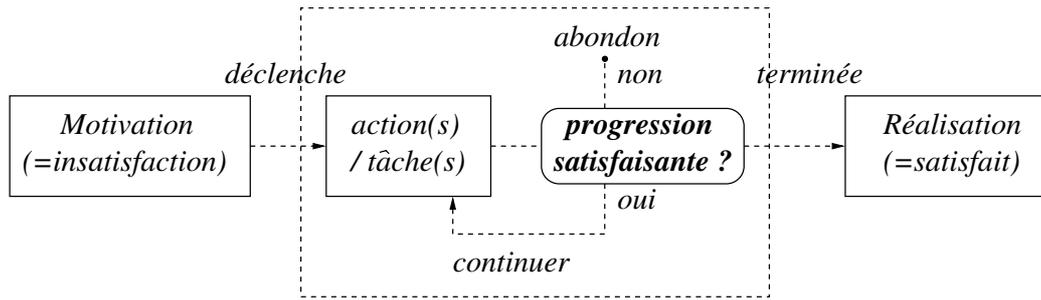


FIG. 4.3 – Fonctionnement d'un agent sur le principe motivation-action-satisfaction

*continue* de la *progression* de la tâche de l'agent. Par contre, notre objectif (initial) n'est pas de connaître la valeur moyenne de l'exécution à la fin de la tâche, mais son état tout au long de l'exécution (cependant, nous montrons au chapitre 10 comment la valeur moyenne de satisfaction de l'agent est employée pour le calcul de la récompense dans un algorithme d'apprentissage par renforcement).

Nous présentons dans la section suivante le calcul de la satisfaction courante de l'agent, qui utilise une mesure de la progression de la tâche s'inspirant des *progress estimators*.

### 4.3 Satisfaction personnelle

Nous considérons un agent seul dans son environnement, ses actions sont dirigées uniquement par des motivations de "survie" et la réalisation de tâches. Ainsi, nous définissons une première satisfaction, liée uniquement à la tâche courante de l'agent, nommée satisfaction personnelle.

**Définition** (Satisfaction personnelle). La satisfaction personnelle d'un agent, à un instant donné, est un état fonction de la progression de sa tâche courante.

Cette définition, volontairement générale, s'appuie sur la notion assez floue de *progression de tâche*. En effet, celle-ci peut exprimer

- soit une progression instantanée de la tâche, prenant en compte l'évolution de la tâche entre *deux instants proches*,
- soit une progression plus globale dans le *temps* liée à la réalisation de la tâche depuis son commencement.

Nous pensons qu'idéalement l'état de satisfaction de l'agent doit représenter ces deux notions à la fois. Ainsi, nous avons proposé une définition du calcul de la satisfaction personnelle basée sur une mesure incrémentale de la progression de la tâche courante (type *progress estimator* de M.J. Mataric, cf. ci-dessus).

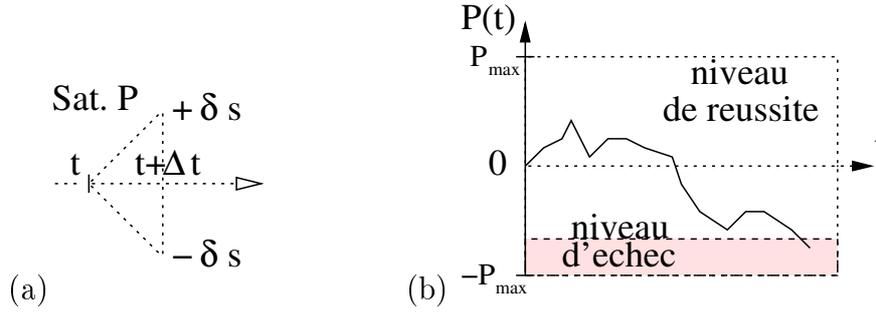


FIG. 4.4 – (a) Évolution de  $P$  pour un pas de temps (b) Exemple de courbe de satisfaction personnelle

### Mesure de la satisfaction personnelle

Notre approche consiste à augmenter ou à diminuer la valeur de la satisfaction en fonction d'une mesure régulière de l'avancement de la sous-tâche courante. La notion de sous-tâche est prise ici dans le sens de comportement/action élémentaire d'une tâche ([Drogoul, 1993], [Arkin, 1998]).

**Définition** (Valeur de satisfaction personnelle). Pour un agent qui exécute une sous-tâche, on définit  $P(t) \in \mathbb{R}$  sa *valeur* de satisfaction personnelle à l'instant  $t$  par

$$- P(t) = P(t - \Delta t) + v ,$$

$v \in \mathbb{R}$  est une mesure de la progression de la sous-tâche sur  $\Delta t$  et  $|v| \leq \delta s$

$$- \forall t \geq 0, |P(t)| \leq P_{max}, P_{max} \text{ et } \delta s \text{ sont des constantes de } \mathbb{R}^+$$

La première condition précise que le niveau de satisfaction de l'agent est une variation, positive ou négative, de la *précédente* valeur de satisfaction (cf. schéma fig. 4.4.a).

La deuxième condition permet de borner la valeur de satisfaction dans l'intervalle  $[-P_{max}, P_{max}]$  (cf. schéma fig. 4.4.b), par conséquent  $\delta s < P_{max}$ . Généralement, on prendra  $\delta s$  petit devant  $P_{max}$ .

Remarque : La discrétisation du temps  $\Delta t$  correspond au cycle perception-délibération-action de l'agent [Ferber, 1995].

**Définition** (*Hédographe*). On appelle Hédographe<sup>1</sup> d'un agent la courbe de satisfaction personnelle construite à partir des valeurs  $P(t)$  obtenues successivement dans le temps.

<sup>1</sup>Cette terminologie est dû à la concaténation des termes *hedos* (du Grec plaisir) et *graphe*.

**Progression instantanée d'une sous-tâche**

L'évolution de  $P$  dépend de  $v$  qui est une mesure de la *progression "instantanée"* de la sous-tâche en cours, celle-ci peut être positive, nulle ou négative. Formellement,

$$v = \begin{cases} m & \text{si progression vers but} \\ n & \text{si éloignement du but} \\ f & \text{si agent immobilisé} \end{cases} \quad (4.2)$$

$$-\delta s < f < n < 0 \leq m < \delta s$$

La différence entre cette mesure et le *progress estimator*  $R_1(t)$  est la spécification d'une pénalisation pour l'immobilisation, qui de plus, est plus forte que celle de l'éloignement :  $f < n < 0$  (le *progress estimator*  $R_2(t)$  ne pénalise pas l'immobilisme forcé). La conséquence étant de provoquer un changement rapide dans le comportement de l'agent pour qu'il sorte d'une telle situation. Ce type de situation pouvant être due aux contraintes de l'environnement ou aux actions de certains agents voisins.

**Satisfaction et comportement**

La valeur  $P(t)$  comme nous l'avons définie est un *niveau de satisfaction acquis dans le temps*, qui intègre les états passés de l'agent. L'intérêt d'une telle approche est de cumuler de l'information pour définir des stratégies comportementales :

- Plus un agent progresse dans sa tâche, plus sa motivation à continuer est grande. Le niveau de satisfaction représente la force de cette motivation (cf. développement de l'architecture chapitre 6).
- Inversement, s'il ne progresse plus, sa motivation décroît au profit d'autres tâches.
- Un agent qui a régulièrement progressé vers son but atteint un niveau de satisfaction élevé, s'il rencontre momentanément des difficultés sa satisfaction chute mais reste positive. Ainsi, avec un niveau de satisfaction élevé, un agent peut accepter plus facilement et plus longtemps des contraintes qui l'éloignent de son but (c'est le principe de recherche d'une meilleure solution en acceptant de sortir temporairement d'un minima local), cf. illustration plus bas.

Remarquons que ce calcul progressif de la satisfaction des sous-tâches permet aussi d'absorber les erreurs éventuelles sur les mesures des capteurs. Nous revenons au chapitre suivant sur les moyens de définir ces mesures.

Pour résumer, la mesure de la satisfaction personnelle permet d'évaluer :

- la *progression courante* des sous-tâches par la variation de  $P$  entre deux instants :  $P(t) - P(t - \Delta t)$  (la dérivée),
- la *qualité de réalisation* des sous-tâches par l'intégration des valeurs  $P(t)$  sur sa période d'exécution,
- les réalisations rapides et efficaces, notées  $P \nearrow^+$ ,
- les comportements/situations d'échecs, notées  $P \searrow^-$  (ex. fig. 4.4.b),
- les actions qui rencontrent des difficultés dans leur réalisation mais qui aboutissent, notées  $P \sim^+$ .

### Illustration : action de coordination spatiale

Considérons des agents mobiles capables de se déplacer en évitant les obstacles qu'ils perçoivent devant eux. Ces évitements sont effectués en contournant les obstacles par le côté qu'ils perçoivent comme le plus "libre" (agents notés *type 1*). Cette technique est efficace pour contourner des obstacles fixes, mais elle peut mener à des blocages ou des ralentissements lorsque les agents se font obstacles [Arkin, 1998].

Prenons une situation simple impliquant trois agents de *type 1* qui évoluent à vitesse égale dans un couloir. Ces agents ont pour seule tâche de traverser le couloir. Deux d'entre eux avancent de front vers le troisième qui arrive dans le sens opposé (cf. fig. 4.5 état initial). Pour étudier le comportement de ces "robots", nous avons réalisé des simulations<sup>2</sup> de cette situation et tracé progressivement l'hédographe de chaque agent. La variation de satisfaction personnelle dépend de l'angle  $\alpha$  entre la direction de déplacement de l'agent et la direction de son but :

$$v = \begin{cases} 0.5 + \cos(\alpha) & \text{si progression vers but} \\ \cos(\alpha) & \text{si éloignement du but} \\ -1.5 & \text{si immobilisé} \end{cases}$$

Ce calcul vérifie bien  $-\delta s < f < n < 0 \leq m < \delta s$  puisque

$\forall \alpha$  tq.  $\frac{\pi}{2} < \alpha < \frac{3\pi}{2}$  (éloignement) alors  $-1.5 < \cos(\alpha) < 0$

et

$\forall \alpha$  tq.  $\frac{\pi}{2} > \alpha$  ou  $\alpha > \frac{3\pi}{2}$  (progression) alors  $0 \leq 0.5 + \cos(\alpha) \leq 1.5$

---

<sup>2</sup>Nous avons développé un simulateur d'agents mobiles à l'aide de la plate-forme MadKit. Ces travaux sont présentés dans les chapitres 8 et 9.

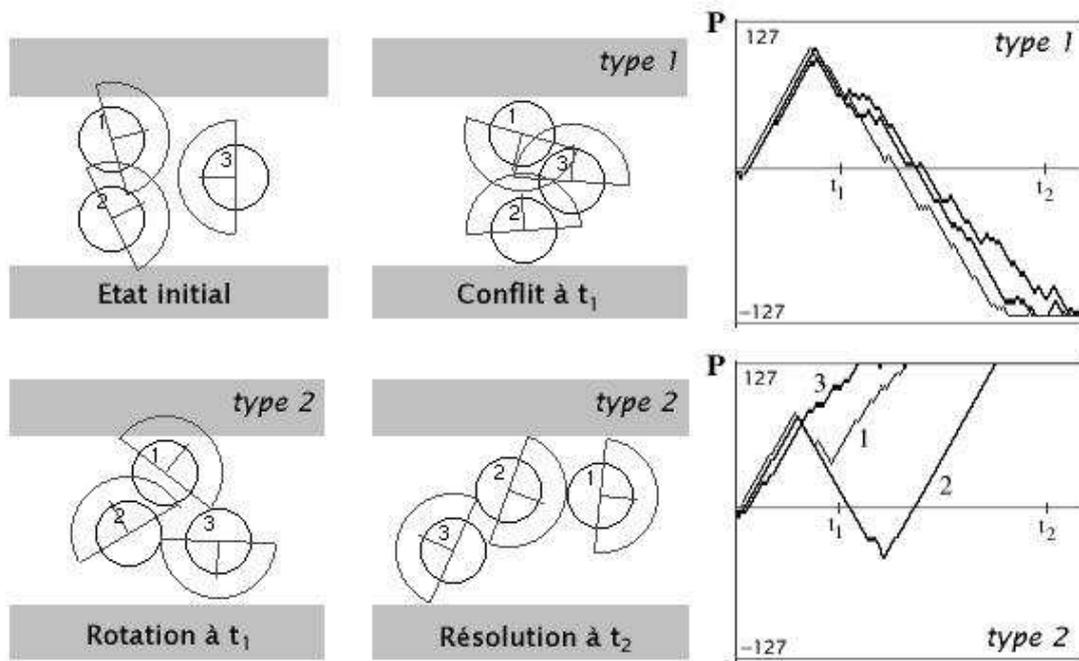


FIG. 4.5 – Copies d'écrans des simulations avec tracés des hédographe superposés des agents, de type 1 : évitement simple des obstacles et de type 2 : intègre le contournement des agents par la gauche.

Ainsi la valeur de  $P$  progresse fortement quand l'agent avance vers son but, elle régresse faiblement lorsque l'agent s'en éloigne et elle baisse fortement si l'agent se trouve immobilisé/bloqué. Dans les simulation, nous avons pris  $P_{max} = +127$ .

Lorsque les trois agents se rencontrent, le couloir est trop étroit pour qu'ils puissent se croiser facilement. Les agents de *types 1* sont incapables de se coordonner pour parvenir à se croiser. Les tracés d'hédographe montrent clairement une chute régulière des satisfactions personnelles jusqu'à la valeur minimum (cf. fig. 4.5).

Modifions le module d'évitement pour que les agents puissent détecter la présence de gêneurs, et qu'ils les contournent toujours *par le même côté* (par exemple à gauche). Nous avons vérifié que cette action était profitable en réalisant de nouvelles simulations (agents notés *type 2*).

Les tracés d'hédographe montrent qu'avec l'action de coordination, la satisfaction ne chute que temporairement lorsque les agents opèrent un croisement par "rotation à gauche". Plus précisément, cela nous permet d'évaluer la résolution : l'agent 3 trouve rapidement un passage (sa satisfaction continue à progresser), l'agent 1 subit une baisse légère avant d'exploiter l'espace libéré par l'agent 3. Enfin, l'agent 2 est plus

longuement gêné par ce processus de rotation, sa satisfaction chute jusqu'à des valeurs légèrement négatives avant de remonter vers le maximum dès que l'agent 1 peut progresser.

#### **Vers l'interaction satisfaisante...**

Le chapitre suivant définit une nouvelle satisfaction de l'agent, liée à ses *interactions* avec son voisinage. Celle-ci permet de compléter la satisfaction personnelle par une vision moins égocentrique des actions. Ainsi, nous en déduisons les notions d'interactions insatisfaisantes ou potentiellement satisfaisante, d'émission *d'influences* et de réaction *altruiste* à ces influences.



# Chapitre 5

## Signaux de satisfaction interactive et réaction altruiste

### 5.1 Objectif

Nous avons vu au chapitre précédent que le simple principe { insatisfaction  $\Rightarrow$  action  $\Rightarrow$  satisfaction } permettait à un agent de se maintenir “en vie”, et pour des entités artificielles de déclencher les tâches pour lesquelles elles étaient prévues. De plus, le modèle de satisfaction personnelle permet à un agent d’évaluer et contrôler la progression de ses tâches. Mais la véritable difficulté, dans la conception d’un SMA, se situe dans la gestion des nombreuses interactions entre agents.

Notre objectif peut maintenant s’exprimer en terme de satisfaction :

---

Soit un système homogène ou hétérogène d’agents situés, *l’objectif* est de

- maintenir ou augmenter l’état de satisfaction personnelle de chaque agent,
  - assurer des interactions satisfaisantes aux agents (coopération, évitement des conflits).
- 

#### Perception des autres...

La satisfaction personnelle modélise un état dépendant de la progression des actions menées par l’agent - sans tenir compte de son voisinage -. Si nous définissons le comportement des agents à partir de cette seule satisfaction personnelle, le résultat sera un ensemble d’agents très “égoïstes”, incapables d’actions intentionnellement coopératives.

Notre but est de développer des agents coopératifs, donc capables de tenir compte des états et des actions des autres agents. Ce qui revient à définir un comportement d'agent tel *qu'il agisse pour se satisfaire mais aussi pour satisfaire les autres*.

Une première idée évidente consiste à donner la possibilité aux agents de *communiquer leur satisfaction personnelle*. Ainsi, chaque agent peut connaître l'état courant de ses voisins. Mais reste le point essentiel, *l'agent doit savoir réagir à cette information, c'est-à-dire savoir agir pour satisfaire l'autre (ou annuler son insatisfaction) (problème 1)*. Or, la satisfaction personnelle ne comporte aucune information sur la nature des besoins de l'agent. Donc, un agent recevant une information de satisfaction personnelle sait seulement que son voisin est satisfait ou insatisfait (nous verrons au chapitre 10 que cela est suffisant pour concevoir des agents capables d'apprendre à coopérer).

Dans le cadre de l'établissement d'interactions satisfaisantes entre agents (type actions coopératives), la seule communication de la satisfaction personnelle n'est pas suffisante. Nous examinons alors les informations que les agents doivent se communiquer pour réaliser des comportements coopératifs.

Nous avons déterminé au chapitre 3 un ensemble de situations insatisfaisantes pour les agents (besoin d'aide, conflits), ainsi que des situations pouvant être améliorées par coopération (partage d'informations, de ressources, évitement de conflits). Nous en avons déduit un tableau de *situation–actions coopératives* (tableau 3.2).

Par ailleurs, nous avons montré l'intérêt de placer le traitement des conflits/problèmes spatiaux en priorité dans le comportement d'un agent.

Ainsi, nous avons reformulé le tableau 3.2 en terme *d'actions spatiales* dont nous précisons pour chacune l'information minimale à communiquer pour la déclencher.

#### 1. Réduire les insatisfactions

type d'insatisfaction	besoin de l'agent	message nécessaire
être gêné par un voisin	départ du gêneur	demande d'éloignement
besoin d'aide	venue d'un agent	appel

#### 2. Augmenter la satisfaction d'autrui

type de coopération	résultat attendu	message nécessaire
partager une ressource	venue d'agents	appel
éviter un conflit, éloigner des agents	repousser les agents	demande d'éloignement

A l'exception de l'échange d'informations complexes (sous forme symbolique), ces tableaux expriment toutes les insatisfactions/coopérations en un mécanisme utilisant seulement les notions *d'appel et de répulsion* entre agents.

### 5.1.1 Évaluer les situations nécessitant des actions coopératives

Nous exprimons les situations nécessitant des actions coopératives, données par la première colonne des deux tableaux précédents, en conditions plus formelles :

#### Satisfaction interactive

La satisfaction interactive est définie par analogie avec la satisfaction personnelle comme une mesure de la qualité d'une interaction (cf. tableau insatisfactions ci-dessus). Plus formellement,

**Définition** (*Satisfaction interactive*). La satisfaction interactive d'un agent A par rapport à un agent B est la *perception de l'incidence* des actions de B sur la tâche en cours de A. Celle-ci appartient à l'une des trois catégories suivantes :

- incidence nulle,
- incidence positive : l'agent perçoit l'interaction comme une aide,
- incidence négative : l'agent perçoit l'interaction comme une gêne.

Conséquence : si une situation d'interaction est insatisfaisante (*incidence négative :  $Sit_1$* ), l'agent déclenchera une demande de *répulsion*.

Remarque : Comme pour la satisfaction personnelle cette évaluation n'est réalisable que si l'agent est muni des "percepts" nécessaires. Nous avons simulé et implémenté ce type de mesure pour la gestion des gênes spatiales entre robots mobiles. L'émission par les robots de signaux infrarouges permet de déterminer facilement la présence d'un robot gêneur dans la direction de déplacement (cf. développement chapitre 12).

#### Conflit potentiel

**Définition** (*Conflit potentiel*). Perception par un agent d'un conflit imminent ou d'une situation pouvant mener à une satisfaction interactive négative.

Conséquence : si un conflit potentiel est détecté ( *$Sit_2$* ), l'agent déclenchera une demande de *répulsion*.

### Situation d'appel

**Définition** (*Situation d'appel*). Etat d'un agent appartenant à l'une des catégories suivantes :

- $Sit_3$  : besoin d'aide,
- $Sit_4$  : connaissance d'une ressource abondante.

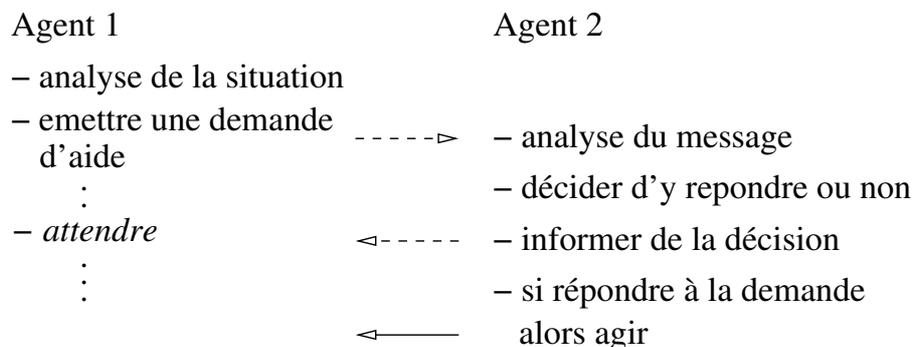
Conséquence : si l'agent est dans l'une des situations  $Sit_3$  ou  $Sit_4$  il déclenchera une demande *d'appel*.

Nous développons ci-après le problème de la communication des demandes (appel, répulsion) et les processus à mettre en oeuvre pour ceux qui les reçoivent.

### 5.1.2 Un mécanisme de communication et de réaction temps réel

#### Schéma classique de communication

Examinons le processus classique mis en oeuvre pour faire interagir deux agents/robots dans un but coopératif (par exemple dans le protocole de coopération de M. P. Singh [Singh, 1998] ou les échanges de messages "sign-board" de Jing Wang [Wang, 1994]) :

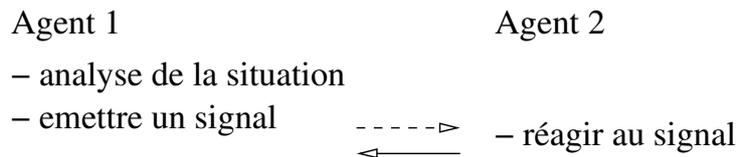


Défauts de l'approche par communication de haut niveau :

1. *délai important (problème 2)* pour que l'agent 1 code la demande, puis que l'agent 2 analyse le message, qu'il émette un nouveau message (parfois qu'il attende une confirmation) et enfin qu'il agisse en réponse au message initial.
2. la réception simultanée de *plusieurs messages* de ce type rend plus complexes et plus longues les opérations de décisions et d'échanges des messages (*problème 3*).

Notre solution consiste à simplifier ce processus en le remplaçant par *la diffusion de signaux locaux* simples à coder et à interpréter.

### Schéma de communication/réaction par signaux



Il s'agit de réduire l'opération d'envoi d'un message par l'agent 1 à un simple *signal* et les opérations de l'agent 2 à une simple *réaction* (*réponse au problème 1*). Les deux intérêts majeurs sont le gain en temps et une plus grande facilité à combiner plusieurs requêtes (*réponses aux problèmes 2 et 3*).

Bien sûr, un tel modèle ne permet pas d'établir des conversations entre agents et la sémantique des échanges est très limitée. Toutefois, cela suffit à coder des demandes d'appels et d'éloignements avec différentes intensités.

Le modèle de coopération "réactive" que nous proposons se décompose en trois phases :

1. déterminer la situation nécessitant des interactions/signaux,
2. modéliser une demande par une simple information pour la traduire en un signal,
3. établir un mécanisme de réaction à ces signaux assurant une réponse rapide aux demandes.

## 5.2 Signaux de satisfaction interactive

Les messages d'attractions ou de répulsions sont nommés signaux de satisfaction interactive car ils représentent une capacité à améliorer la satisfaction des agents en modifiant leur interaction. Formellement :

**Définition** (*Signal de satisfaction interactive*). Le signal de satisfaction interactive d'un agent A vers un agent B est une valeur  $I \in [-P_{max}, P_{max}]$  qui exprime une demande de A :

- $I < 0$  pour repousser l'agent B (émis en situations  $Sit_1$  ou  $Sit_2$ ),
- $I > 0$  pour attirer l'agent B (émis en situations  $Sit_3$  ou  $Sit_4$ ).

Un signal de satisfaction interactive permet d'exprimer une demande très simple, une attraction ou une répulsion, et de donner **une force à cette demande (valeur absolue de  $I$ )**. La dernière phase du processus consiste à définir le comportement de l'agent lorsqu'il reçoit un signal de satisfaction interactive, c'est à dire sa réaction.

### 5.3 Réaction altruiste

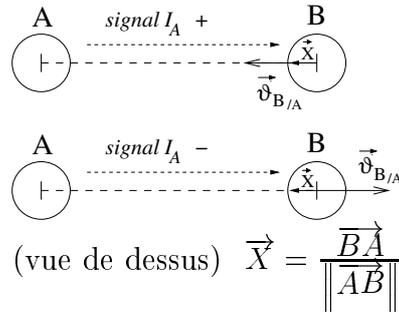
Les agents doivent réagir en temps réel aux signaux émis (déplacements dans un environnement inconnu et dynamique). Ainsi, nous avons adopté le modèle le plus efficace en terme de navigation autonome. Il s'agit du modèle vectoriel, permettant de combiner efficacement différentes contraintes imposées à l'agent.

#### Définition du vecteur altruisme

Nous supposons ici que l'agent répond systématiquement aux signaux qu'il reçoit. Nous parlons alors d'agents "altruistes", cette notion est développée dans le chapitre suivant.

Nous posons le *vecteur altruisme*, définissant la réaction altruiste d'un agent (B) à un signal de satisfaction interactive émis par un agent (A) :

$$\overrightarrow{\vartheta}_{B/A}(t) = k \cdot \text{Sign}(I_A(t)) \cdot |I_A(t)| \cdot \overrightarrow{X} \quad (5.1)$$



L'équation 5.1, où  $k$  est un gain, définit un champ de force (de la forme champ attractif/répulsif). La réaction  $\overrightarrow{\vartheta}_{B/A}(t)$  est un vecteur dont le support est la droite  $AB$ , la norme est proportionnelle à la valeur absolue de l'information transmise  $|I_A(t)|$  et le sens est donné par le signe de  $I_A(t)$ .

Nous avons exprimé le signe de la réaction par une fonction  $\text{Sign}$  de la valeur perçue. La raison est que suivant le codage utilisé pour transmettre l'information, l'évaluation de son signe peut nécessiter une opération.

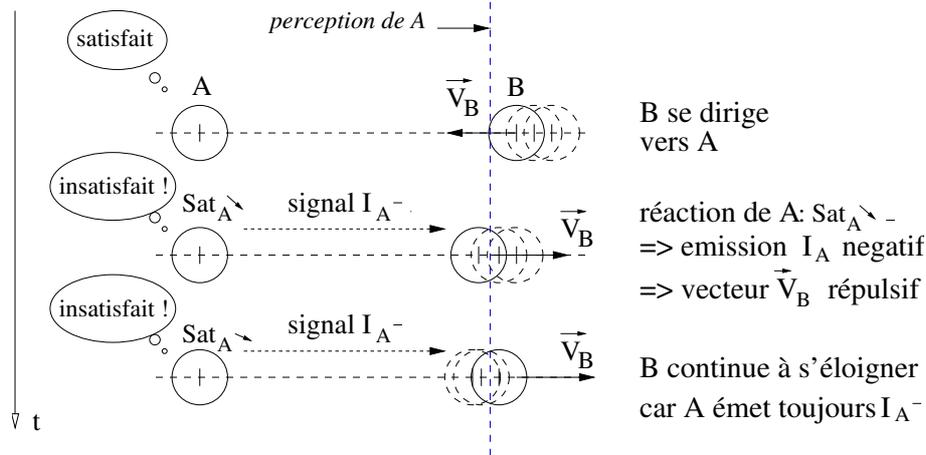


FIG. 5.1 – Illustration d’une réaction altruiste à un signal de répulsion

Ainsi,

- un signal positif construit un vecteur dont le sens est celui de  $\vec{X}$ , donc il s’agit bien d’une attraction vers A,
- un signal négatif construit un vecteur dont le sens est opposé à celui de  $\vec{X}$ , donc il s’agit bien d’une répulsion de A.

Le vecteur altruisme peut être aussi construit en tenant compte de la distance entre l’émetteur et le récepteur, comme dans les champs de potentiels (nous utilisons cette variante dans le traitement du problème des robots fourrageurs, cf. chapitre 8). Pour cela, nous tenons compte d’une puissance  $n \geq 1$  de la norme  $\|\vec{AB}\|$  :

$$\vec{v}_{B/A}(t) = k \cdot \text{Sign}(I_A(t)) \cdot |I_A(t)| \cdot \frac{\vec{X}}{\|\vec{AB}\|^n} \quad (5.2)$$

Ainsi, plus les agents sont proches, plus forte est l’attraction ou la répulsion. Cette distance est supposée évaluée par la perception de l’intensité des signaux reçus (type infra-rouges, cf. réalisation chapitre 12).

### Illustration

La figure 5.1 illustre le fonctionnement du vecteur altruisme pour le cas où un agent (A) est gêné par la présence d’un autre (B) (formule 5.1).

1. Lorsque l’agent B entre dans la zone de perception de A, cela déclenche une insatisfaction interactive de A,

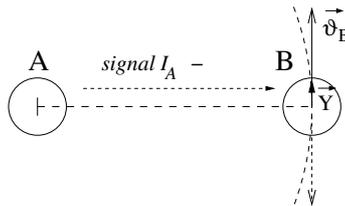
2. En conséquence A émet un signal répulsif qui est perçu par B, celui-ci applique pour trajectoire le vecteur altruisme,
3. B s'éloigne de A jusqu'à sortir du champ de perception de A.

Ici, l'agent B n'émet pas de signaux, donc il n'y a pas de réaction directe de la part de l'agent A, si ce n'est la prise en compte de l'évolution géographique de B (perception du mouvement). Le chapitre suivant est consacré à l'application de ce principe de réaction dans un système où *plusieurs agents* émettent *simultanément des signaux* de satisfaction interactive.

### Contourner plutôt que de fuir...

Dans le cas de la réception d'un signal répulsif, nous avons posé une troisième variante de la réaction altruiste. K. Zeghal a montré que l'évitement de plusieurs obstacles par un agent mobile pouvait être amélioré par l'introduction d'une force de glissement (ou de contournement) [Zeghal, 1994]. Cette force de glissement permet d'éviter l'existence d'attracteurs tels les équilibres stables et les cycles limites. Ainsi, nous utilisons cette force pour définir une deuxième formule de la réaction altruiste (en cas de répulsion) :

$$\begin{aligned} &\text{si } \text{Sign}(I_A(t)) < 0 \text{ alors} \\ \overrightarrow{\vartheta}_{B/A}'(t) &= \pm k \cdot |I_A(t)| \cdot \vec{Y} \end{aligned} \quad (5.3)$$



où  $\vec{Y}$  est un vecteur unitaire normal à la droite  $(AB)$  et d'origine  $B$ .

#### 5.3.1 Nature des signaux

Nous avons supposé jusqu'à présent que les agents pouvaient transmettre à leur voisinage des valeurs signées. Le principe essentiel de ces communications étant d'influencer un ou plusieurs agents simultanément dans un rayon limité. Ceci peut se représenter par la diffusion autour d'un agent d'un signal concentrique (fig. 5.2).

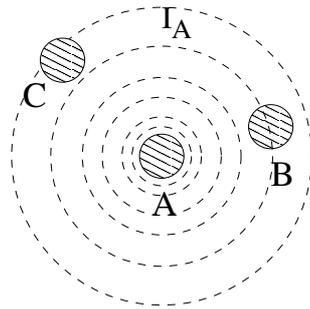


FIG. 5.2 – Représentation de l'émission d'un signal de satisfaction interactive d'un agent vers ses voisins

Nous précisons ici les paramètres définissant ces signaux et nous évoquons les moyens de les mettre en oeuvre.

Les signaux sont définis par deux caractéristiques principales :

- la *valeur signée* qu'ils transportent,
- la *distance* à laquelle ils peuvent être perçus.

Généralement, l'échange de messages (symboles) peut se faire par communication directe point à point ou par diffusion dans un réseau. Les signaux, à l'inverse, ont un mode de communication uniquement par diffusion en se propageant dans l'environnement. Ils peuvent être de diverses natures : ondes (sonores, lumineuses, IR, électromagnétique), chimique, tactile ou électriques. Les trois dernières étant quasi exclusivement utilisées par les animaux et les insectes.

La nécessité d'émettre *localement* une information de faible contenu nous a fait choisir une solution à base de signaux d'ondes s'atténuant sur de courtes distances (IR). Nous avons envisagé de coder les valeurs *signées* de diverses manières. Ces méthodes, qui ne fournissent pas les mêmes avantages, sont présentées ci-dessous :

1. La valeur est codée par une *variation d'intensité* du signal dans le temps : l'amplitude de la variation définit sa valeur absolue et le sens de la variation son signe. Inconvénients : la lecture d'une variation d'amplitude reste approximative, cette variation est consommatrice en temps et elle lie nécessairement valeur et distance émetteur-récepteur. Mais surtout, il y a systématiquement interférence des signaux en présence de plusieurs agents communicants simultanément.
2. La valeur est codée par une fréquence sur une échelle représentant l'intervalle  $[-P_{max}, P_{max}]$ . Cette solution exige en réception une écoute de toutes les fréquences.

De plus il n'est pas possible de distinguer plusieurs réceptions à la même fréquence/valeur. Enfin, l'approche est coûteuse, avec des risques d'interférences importants et nécessite donc l'emploi d'un protocole.

3. la valeur est codée par quelques bits émis sur une fréquence partagée par tous les agents. Cette solution nécessite un protocole pour éviter les interférences de trames.

Nous avons retenu la troisième approche, que nous avons développé sous forme d'un protocole dédié aux systèmes multi-agents situés, cf. chapitres 11 et 12.

## 5.4 vers la satisfaction du système...

Les signaux de satisfactions permettent aux agents de s'échanger rapidement des requêtes, que nous nommerons influences, dans le but de résoudre des *conflits* et de *s'entre-aider*. L'agent dispose maintenant d'une satisfaction personnelle centrée sur ses propres actions, d'une perception des actions extérieures et de signaux lui permettant d'essayer d'améliorer sa situation et/ou celle des autres.

La troisième partie de la thèse présente l'intégration des satisfactions et du mécanisme de réaction altruiste dans une architecture d'agent. Le but étant de faire coopérer, en temps réel, de nombreux agents situés.

## Troisième partie

### Le modèle Satisfaction-Altruisme



# Chapitre 6

## Coopération par altruisme et signaux de satisfaction

### 6.1 Comportements altruistes

#### Chez les animaux



De nombreux animaux et insectes présentent des comportements altruistes. David McFarland définit l'altruisme comme un comportement autodestructeur profitant aux autres individus [McFarland, 1987]. "On peut évaluer son coût pour l'altruiste et le profit de celui qui le reçoit en terme de valeur adaptative". En effet, ces comportements sont particulièrement présents entre individus ayant une parenté proche. L'altruisme se manifeste par des activités de soins, de nourrissage ou de protection des parents envers la progéniture. L'intérêt des parents dans ces comportements est d'assurer une plus grande chance de survie aux jeunes, donc d'augmenter les chances de copie de leurs propres gènes.

On trouve aussi des comportements altruistes entre animaux d'espèces différentes. Dans ce cas, on parle d'altruisme réciproque, qui est défini comme une forme de *symbiose* avec un *délai* entre le geste altruiste et sa récompense.

Encore une fois, ce sont les insectes sociaux qui ont poussé la notion d'altruisme à sa conclusion logique extrême. Celle de pouvoir se sacrifier pour le bien de la communauté

(par exemple l'abeille lorsqu'elle pique un agresseur de la ruche, ou encore les attaques suicidaires des fourmis ou des termites sur les pillards de nids cf. [McFarland, 1987]).

### Pour des robots...

Dans le cadre de systèmes artificiels, composés de plusieurs robots/agents travaillant en “sociétés”, il ne s'agit pas de “sacrifier” des robots au profit de la réalisation des tâches. Mais il est intéressant de reprendre le principe d'un comportement avant tout bénéfique pour la communauté.

Les agents ou les robots que nous considérons ne sont pas capables de se reproduire, leur activité se borne à effectuer le mieux possible un ensemble de tâches. Cependant, la problématique de la survie existe dans les deux types de systèmes. Des robots censés évoluer dans un environnement inconnu et hostile peuvent être confrontés à des problèmes de pannes, de manque d'énergie, de blocages, voire d'agressions par d'autres robots (en cas de compétition).

Un système composé de plusieurs agents situés - comme des robots mobiles - n'a pas les mêmes objectifs ni les mêmes besoins qu'une meute d'animaux ou qu'une société d'insectes. Cependant, de fortes similitudes existent :

- chaque agent doit s'efforcer de rester “vivant” (pour un robot : éviter les chocs et la panne d'énergie),
- chaque agent réalise de *nombreux déplacements* parmi les autres agents,
- chaque agent doit remplir des tâches individuelles,
- les agents peuvent *s'entre-aider* pour réaliser des tâches.

Notre approche consiste à mettre en place des interactions “altruistes” entre agents situés individuels.

## 6.2 Principe de coopération

Le principe de traitement que nous adoptons est celui de la résolution collective de problème [Bonabeau and Theraulaz, 1994] [Drogoul, 1993] [Ferber, 1995], qui s'appuie en particulier sur *l'auto-organisation* des agents. Afin d'accélérer certains processus, nous introduisons des échanges de signaux (ou influences) entre agents qui permettent (i) d'éviter et résoudre certains conflits (cf. chapitre 9) et (ii) de coopérer (recrutement, aide, coordination des trajectoires) (chapitre 8).

**Principe de fonctionnement proposé pour un système multi-agents situés**

- Au niveau *local*, chaque agent :

1. tente d'éviter/résoudre les conflits locaux,
2. tente de répondre aux appels,
3. partage les ressources abondantes,
4. effectue des tâches pré-codées.

Cet ordre de priorité, introduit en section 3.3.2.1, est justifié au cours de la thèse.

- Au niveau *global* : la réduction des insatisfactions locales améliore l'état général du système, il s'auto-organise pour être satisfait (cf. développement chapitres 8 et 9).

Toute notre approche repose sur une bonne gestion des interactions locales. Nous avons explicité dans les chapitres précédents les situations insatisfaisantes ou pouvant être améliorées par des comportements coopératifs. Nous généralisons la notion de réaction altruiste à des *interactions altruistes* mettant en jeu plusieurs agents voisins. Nous traitons d'abord le problème entre deux agents, puis nous donnons une solution générale pour  $n$  agents.

## 6.3 Des interactions satisfaisantes

Hypothèses sur les agents, ils sont capables :

- d'effectuer des actions/tâches individuelles (données par le concepteur),
- d'évaluer leur satisfaction personnelle, interactive et les situations nécessitant des appels,
- d'émettre des signaux d'attraction ou de répulsion,
- de réagir par construction du vecteur altruisme.

### Deux agents - deux émissions

Le tableau 6.1 donne le comportement de deux agents A et B exécutant le vecteur altruisme suivant les signes de  $I_A(t)$  et  $I_B(t)$  :

Il se dégage de ce tableau que les deux interactions mettant en jeu des demandes de même signe sont correctement gérées (lignes 1 et 4), les deux autres provoquent une situation insoluble (lignes 2 et 3).

	$Sign(I_A)$	$Sign(I_B)$	Situation	interaction	comport. de A et B
1	–	–	gènes mutuelles	conflit	évitements mutuels
2	–	+	intérêts incompatibles	conflit/appel	dépl. identiques
3	+	–	intérêts incompatibles	appel/conflit	dépl. identiques
4	+	+	appels mutuels	coopération	rapprochement mut.

TAB. 6.1 – Interaction entre deux agents émettant chacun un signal de satisfaction interactive

En effet, deux gènes mutuelles provoquent des répulsions opposées, ce qui satisfait chaque agent. Inversement, deux agents s'appelant mutuellement vont être amenés à se rapprocher. Mais ce comportement peut ne pas être satisfaisant lorsqu'un agent veut attirer en un lieu précis un autre agent.

Les deux autres cas correspondent à des situations où les demandes sont opposées : l'un des agents est appelé par l'autre tout en le repoussant. Le résultat est une situation où les deux agents se déplacent dans le même sens, c'est-à-dire de l'appelé vers l'appelant, mais sans jamais modifier leur distance relative ! Ce qui n'est satisfaisant pour aucun des deux agents.

Ces problèmes peuvent être résolus par la détermination d'une priorité entre les valeurs. Nous posons ici une priorité aux signaux les plus intenses :

**Principe** (*de priorité aux signaux les plus intenses*) : Entre deux signaux de satisfaction, celui de plus grande valeur absolue est prioritaire.

Ce qui peut s'écrire par la fonction  $prioInt(I_1, I_2, \dots, I_n) = \{ I_j \text{ tq. } \forall I_{i \neq j} |I_j| \geq |I_i| \}$

Si la fonction fournit un ensemble de signaux d'intensité équivalente et prioritaire, il faut appliquer une nouvelle fonction de sélection (cf. plus bas).

Ce choix de priorité est justifié par l'hypothèse que les agents émettent des signaux suivant l'importance ou l'urgence de leurs besoins. En particulier, nous verrons que des signaux répulsifs très forts peuvent exprimer un conflit majeur et des signaux attractifs très forts un besoin d'aide urgent.

Nous verrons au chapitre 9 qu'en développant ce principe de priorité il est possible de résoudre des conflits de navigation entre de nombreux agents.

Pour appliquer ce principe aux agents altruistes, nous modifions leur comportement de telle sorte qu'ils n'appliquent le vecteur altruisme que si le signal qu'ils perçoivent est prioritaire sur le leur (les signaux sont supposés être d'intensités différentes).

Ainsi, nous obtenons un nouveau tableau des interactions (tab. 6.2). Il ne contient plus

	$Sign(I_A)$	$Sign(I_B)$	Situation	interaction	comp. de A et B
1	–	–	gènes mutuelles	conflit	éloignement de A ou B
2	–	+	intérêts incompatibles	conflit/appel	B s'éloigne ou A s'appr.
3	+	–	intérêts incompatibles	appel/conflit	A s'éloigne ou B. s'appr.
4	+	+	appels mutuels	coopération	A ou B se rapproche

TAB. 6.2 – Interaction entre deux agents émettant chacun un signal de satisfaction interactive (signaux d'intensités différentes)

de comportements aberrants, puisque seulement un des deux agents en interaction peut réagir. Notons cependant que dans le cas des lignes 2 et 3, seul l'un des deux agents est satisfait.

Si les signaux échangés sont d'intensités équivalentes, nous posons que les agents réagissent seulement si leurs valeurs sont de même signe (deux attractions ou deux répulsions). Dans le cas contraire ils s'ignorent et ainsi ne se déplacent pas (ceci évite la poursuite sans fin).

### Le problème : interaction à $n$ agents - $n$ émissions

Considérons un système occupé par des agents pouvant émettre des signaux d'attractions et de répulsions. Chaque agent évolue ainsi dans un univers dynamique *de contraintes et de demandes* (obstacles physiques, signaux d'appels, de répulsions). Pourtant, chaque agent doit suivre un comportement cohérent vis-à-vis de la réalisation de ses tâches et des actions coopératives possibles.

Comme tout être vivant, un agent artificiel ne peut agir pour réaliser plusieurs tâches incompatibles à la fois ou répondre simultanément à plusieurs demandes indépendantes. Il doit sélectionner une activité principale. En même temps, il arrive que plusieurs contraintes puissent être prises en compte simultanément. C'est le cas pour le calcul d'une trajectoire, où l'on tient compte d'un ensemble d'obstacles. On parle alors de combinaison des contraintes.

La solution que nous proposons à ce problème de sélection/comboinaison est le *modèle satisfaction-altruisme* qui donne à l'agent un comportement lui permettant de sélectionner en temps réel une tâche cohérente vis-à-vis de ses voisins et de combiner plusieurs contraintes lorsque cela est possible.

## 6.4 Le modèle satisfaction-altruisme

### 6.4.1 Sélection d'une action ou tâche (en temps réel)

Ce qui détermine la cohérence du comportement d'un agent c'est avant tout sa capacité à *bien choisir* les tâches à mener, et dans un deuxième temps, à être capable de *persister* un certain temps dans leur accomplissement.

Notre objectif étant la définition d'agents coopératifs, le choix d'une tâche à suivre se fait donc en premier lieu entre une action personnelle et une action coopérative. Ces deux types de tâches étant souvent incompatibles.

Le mécanisme que nous proposons pour effectuer ce choix s'appuie directement sur le modèle des satisfactions présenté dans les chapitres précédents. En effet, les signaux de satisfaction interactive représentent la force des demandes extérieures et la satisfaction personnelle représente la force à continuer une tâche individuelle.

La satisfaction personnelle et les signaux de satisfaction interactive sont définis par une valeur appartenant à l'intervalle  $[-P_{max}, P_{max}]$  (cf. définitions sections 4.3 et 5.2). Ainsi, nous pouvons déterminer le déclenchement d'une tâche coopérative par le test - dit *d'altruisme* - suivant :

---


$$Si \quad \alpha \cdot |I_{ext}(t)| > (1 - \alpha) \cdot P(t) \quad \text{alors} \quad (6.1)$$

arrêter tâche en cours et répondre au signal  $I_{ext}$

---

$\alpha$  est le *coefficient d'altruisme* de l'agent ( $\alpha \in [0, 1]$ )

Supposons  $\alpha = 0.5$  : L'agent ne répondra à un signal extérieur que si celui-ci a une valeur absolue plus grande que sa satisfaction personnelle. Si l'agent est insatisfait ( $P(t) < 0$ ), il abandonne immédiatement sa tâche pour répondre à une demande extérieure. Si l'agent est satisfait ( $P(t) > 0$ ), la demande doit être plus intense que sa propre satisfaction pour y répondre.

Le coefficient  $\alpha$  permet de "jouer" sur l'aspect altruiste de l'agent. On peut lui donner un caractère très altruiste en prenant  $\alpha > 0.5$  et un caractère peu altruiste avec  $\alpha < 0.5$ . L'agent serait purement égoïste pour  $\alpha = 0$  et purement altruiste pour  $\alpha = 1$ .

Enfin, la valeur de réception  $I_{ext}$  peut elle aussi subir une transformation pour la

pondérée suivant la distance à l'émetteur (type champs de potentiels d'obstacles) :

$$I_{ext} = k \cdot \frac{I}{dist^n} \text{ (cf. application chapitre 8)}$$

Nous distinguons

- les tâches individuelles pré-codées déclenchables par la perception de stimuli autres que les signaux de satisfaction interactive,
- les réactions altruistes (ou coopératives) aux signaux de satisfaction interactive.

### Choix de la tâche personnelle

Cette situation se présente lorsque l'agent ne reçoit aucun signal ou bien à la réception de signaux d'intensités inférieures à sa satisfaction personnelle. Dans ce cas, il peut continuer sa tâche courante ou commencer une nouvelle tâche individuelle. Cette décision est prise par comparaison du niveau de satisfaction personnelle (progression de la tâche courante) et des tâches potentiellement exécutables. Le détail de ce processus est donné dans le chapitre suivant.

### Choix de l'action coopérative

L'agent reçoit au moins un signal dont la valeur de satisfaction interactive est plus intense que sa satisfaction personnelle (signaux notés  $I_1, I_2, \dots, I_n$ ). Comme il ne peut répondre qu'à un seul de ces signaux à la fois, nous appliquons le principe de priorité aux signaux les plus intenses. Si plusieurs signaux sont de forces prioritaires égales, l'agent fait un simple choix aléatoire :

$$I_{select} = alea(prioInt(I_1, I_2, \dots, I_n)) \quad (6.2)$$

où  $alea(arg_1, arg_2, \dots)$  est une fonction qui réalise un tirage aléatoire à distribution uniforme sur les valeurs de ses arguments.

La nouvelle tâche de l'agent devient la réponse altruiste qu'il peut faire au signal  $I_{select}$ .

Nous pouvons maintenant aborder l'exécution des tâches et l'émission/réception des signaux.

## 6.4.2 Étude et analyse des systèmes à base de champs de potentiels (forces)

### Avantages et défauts de l'approche

Avantages :

- *l'espace des actions potentielles est infini* (transformation par une fonction mathématique des perceptions en comportement),
- *rapidité des (ré)actions* de part la simplicité des calculs,
- *composition* naturelle des tendances (perceptions, contraintes, but, bruit),
- *anticipation* des conflits  $\Rightarrow$  *fluidité* des mouvements, *résultats performants* pour l'évitement d'obstacles (dynamiques) et la coordination de mouvements entre agents,
- les problèmes de la robotique collective sont essentiellement *des contraintes spatiales*,
- coordination des trajectoires de plusieurs agents homogènes sans échanges de messages.

Inconvénients :

- risques de blocage dans des minima locaux (états d'équilibre entre les forces), de cyclage ou d'oscillation entre des obstacles,
- pas de planification de parcours.

Dans ce type d'approche, l'agent réagit en temps réel à ses perceptions *locales*. C'est cette limitation qui peut le mener à des situations bloquantes ou cycliques (cf. [Koren and Borenstein, 1991]). Ces problèmes ont toutefois été largement traités pour permettre l'emploi de l'approche en navigation autonome (pour des milieux inconnus). Les principales solutions sont l'ajout de bruit pour briser les équilibres, l'utilisation de l'apprentissage pour adapter les paramètres de l'agent à son environnement [Clark et al., 1992] et le calcul de champs de potentiels harmoniques sur une carte de l'environnement [Connolly and Grupen, 1992]. Nous reviendrons sur le traitement des situations pièges aux chapitres 8 et 9.

### Nécessité d'introduire des communications...

Le dernier point des avantages énoncés précise que la coordination entre plusieurs agents peut se faire sans communication (et seulement par perception locale des autres agents). En réalité, cette affirmation a des limites. Il s'agit en général de solutions locales permettant à

des agents identiques d'éviter des conflits de trajectoires par application de règles simples (exemple : toujours se contourner par le même côté). Mais ces solutions peuvent être inefficaces en milieux contraints et dynamiques, cf. [Zeghal, 1994]. Aussi, les solutions par perception locale sont généralement employées pour le maintien en formation d'une équipe d'agents/robots [Balch and Arkin, 1995] et pour la navigation individuelle.

K. Zeghal a montré dans sa thèse, sans l'appliquer, la nécessité d'introduire des communications entre agents pour mettre en oeuvre des interactions efficaces (optimales) si les agents n'ont pas des comportements homogènes [Zeghal, 1994]. En effet, un agent ne peut tenir compte réellement des trajectoires de ses voisins que s'il connaît leurs intentions à court terme. Dans notre cas de figure, nous utilisons un mode de communication - par émission d'un signal à valeur dans  $R$  - pour *ne pas perdre l'aspect réactif du modèle vectoriel*. En effet les intentions des agents sont codées très simplement en signaux et sont décodés/transformés en vecteurs dès leur réception.

### Travaux existants

Rappelons les travaux essentiels concernant l'application des méthodes champs de potentiels / vecteurs aux agents situés. Ce sont des tâches de navigation individuelle ou collective en milieu contraint (obstacles dynamiques ou statique), le maintien en formation d'un groupe d'agents (ou flocking), la réalisation de tâches en environnements inconnus telles que le *foraging* (recherche et transport vers une base d'éléments non localisés, cf. l'étude au chapitre 8), le *consuming* (recherche et consommation de ressources ponctuelles) et le *grazing* (parcours et/ou consommation méthodique d'un environnement-ressource) et enfin du contrôle de robots footballeurs :

- en simulation : O. Khatib [Khatib, 1985] (navigation individuelle, bras manipulateur), C. W. Reynolds [Reynolds, 1987] (simulation de volées d'oiseaux ou "boids"), K. Zeghal [Zeghal, 1994] (coordination de trajectoires, application au contrôle aérien), R.C. Arkin [Arkin, 1992a] (coopération émergente pour le transport),
- avec des robots réels : R. C. Arkin [Arkin, 1987] [Arkin, 1998] (*Schema-Based Behaviors* pour la navigation individuelle et la réalisation de tâches spatiales), M. Mataric [Mataric, 1994a] (groupes de robots, application du modèle *schema-based behaviors* à des tâches spatiales et de déplacements en formation), [Balch and Arkin, 1995] [Arkin and Balch, 1997] (coordination pour le maintien en formation d'équipes de robots).

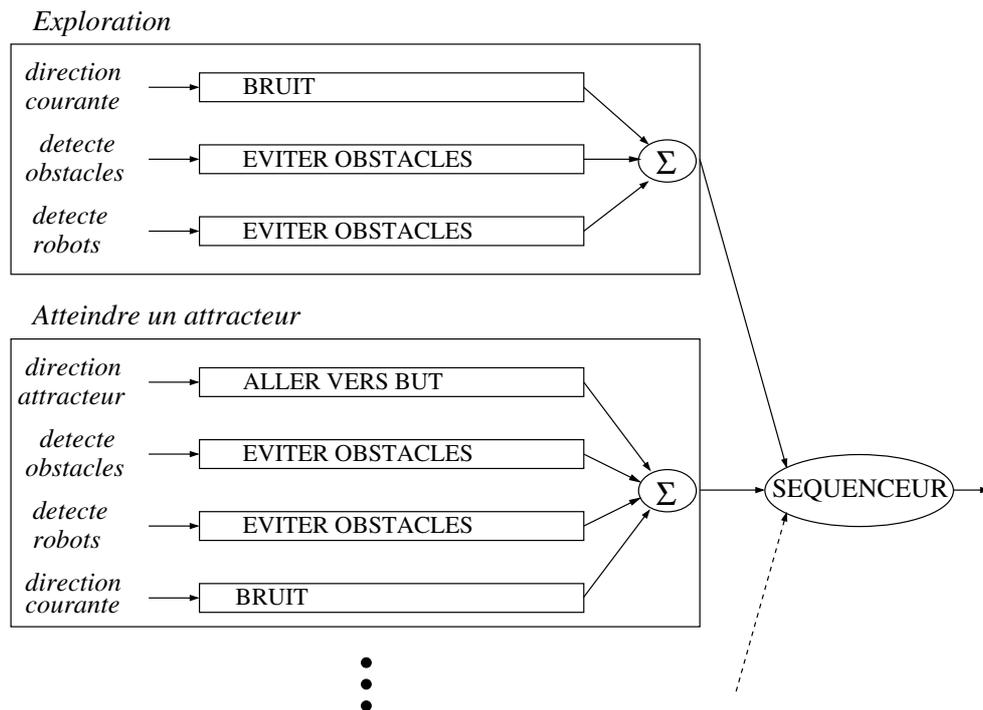
- robots cellulaires de T. Fukuda [Fukuda et al., 1995] : le programme CEBOT qui a abouti à une architecture d’agent dont la navigation est basée sur la sommation de vecteurs/perceptions [Cai et al., 1995] (de type *schema-based behaviors* mais avec un système de hiérarchie pré-définie et adaptatif entre les tâches).
- équipes de robots footballeurs : M. Veloso et P. Stone proposent depuis le milieu des années 90 des architectures de robots footballeurs très performantes. Le principe général étant l’organisation en graphe direct acyclique d’un ensemble de comportements décomposés en primitives. Les agents détiennent un rôle et sont organisés en formations liées à des stratégies [Stone and Veloso, 1999]. En 1998, ils ont introduit un algorithme de sélection d’action permettant aux équipiers non actifs (sans balle) de se positionner pour recevoir une éventuelle passe. Ce processus, basé sur *la combinaison d’attractions* (du porteur de la balle et du but adverse) et *de répulsions* (des adversaires et des équipiers), a rendu leur équipe plus performante [Veloso et al., 1999]. Leurs derniers travaux généralisent cette approche par l’emploi de champs de potentiels pour le contrôle de chaque agent et de l’équipe (maintien en formation et éloignement des bords du terrain ou de “lignes arrières” sont combinés avec les contraintes individuelles) [Stone and McAllester, 2001]. Leurs premiers résultats sont très positifs et nous renforcent dans l’approche “champs de forces” que nous poursuivons depuis trois ans.

Le modèle *Schema-based behavior* [Arkin, 1998] fait référence dans le monde de la navigation individuelle par champs de potentiels (ou de forces). Il a été aussi largement employé pour la coordination de plusieurs agents et le traitement collectif de certaines tâches : (i) maintien en formation de plusieurs robots mobiles par perception locale (ii) auto-organisation de robots dans des tâches de type foraging, consuming et grazing.

Le principe des *schema* de l’architecture de R. C. Arkin est représenté en figure 6.1 et peut se résumer ainsi :

Chaque sous tâche (ou primitive de tâche) nommée *schema* est définie par la composition vectorielle de plusieurs *motor schemas*. Un *motor schema* étant la traduction d’une perception en vecteur direction.

Le séquenceur décide de la sous tâche à suivre (donc de la direction de l’agent) en fonction de la dernière tâche accomplie et des perceptions. Plus généralement le séquenceur suit un diagramme prédéfini des comportements à déclencher.

FIG. 6.1 – Principe des *schema* de l'architecture de R. C. Arkin

A partir d'une telle architecture T. Balch et R.C. Arkin ont mené une étude comparative des performances coopératives d'un système multi-robots sur des tâches de *foraging*, *consuming* et *grazing* suivant différents modes de communications [Balch and Arkin, 1994]. Les agents peuvent s'informer de la présence d'attracteurs suivant les trois modes : - aucune communication directe -, - broadcast d'un bit d'état - et - message contenant la position des buts -. Leurs conclusions sont

1. les communications améliorent significativement les performances des tâches détenant peu de communication indirecte (*forage* et *consume*). Elles ne sont pas nécessaires pour des tâches où l'information est implicite (*graze*).
2. les communications des buts (par messages) n'apportent pas de bénéfices par rapport à la simple diffusion d'un bit d'état (précisant le type de tâche en cours).

Tout d'abord ces travaux confirment que les communications assurent des comportements coopératifs efficaces pour le partage d'informations non inscrites dans l'environnement. Puis, que la diffusion de "signaux" d'états internes est suffisante pour partager des informations pertinentes sur les ressources (notons que c'est une solution qui se limite ici qu'à ce type de coopération). Ceci est cohérent avec notre choix d'émettre une information minimale à l'aide de signaux. Par contre, les informations que nous émettons contiennent

une *demande* liée aux *interactions* des agents.

Depuis 1998, R.C. Arkin a intégré une couche délibérative à son architecture permettant de reconfigurer le comportement du robot, de planifier des tâches et de tenir compte de cartes de l'environnement [Stoytchev and Arkin, 2001]. Mais son approche pour la coopération reste le pré-codage et s'appuie sur l'auto-organisation. De notre point de vue, ce type d'architecture hybride ne propose pas de véritable solution à la coopération intentionnelle en temps réel entre plusieurs agents mobiles individuels.

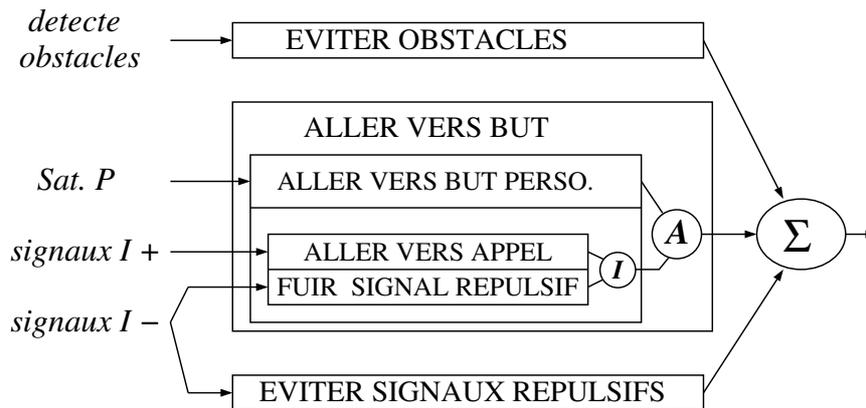
Notre approche, à l'opposée de "l'hybridation", est d'intégrer la coopération au coeur du système de navigation réactif. Ainsi, nous évitons les problèmes de complexité et de fragilité des architectures délibératives. L'emploi de communications *nécessaires*, mais effectuées par signaux, permet l'échange de demandes ou d'intentions coopératives qui sont traitables et exprimables dans le modèle vectoriel.

Nous avons montré au chapitre 3 qu'un système d'agents situés mobiles devait être conçu en priorité autour des problèmes de navigation. Et dans le même temps, que chaque agent devait être capable de remplir certaines tâches lorsqu'il en détectait l'opportunité. Le modèle que nous proposons intègre le processus de sélection de tâche au coeur du système de navigation.

### 6.4.3 Intégration de la sélection de tâche et les actions coopératives au niveau vectoriel

Les réactions altruistes des agents sont intégrées au sein même de la combinaison vectorielle définissant les actions-déplacements. C'est-à-dire qu'une réponse altruiste à un signal est un abandon de la tâche courante pour un nouveau déplacement tendant à satisfaire son émetteur. Il peut s'agir d'une réponse à un appel ou à une répulsion, qui devient alors le but de l'agent. Les signaux répulsifs sont doublement utilisés dans la mesure où ils sont systématiquement intégrés au calcul de la trajectoire. En effet, les contraintes pour l'agent proviennent des obstacles "perçus" et des agents exigeant de "l'espace".

Nous donnons une première expression du modèle satisfaction-altruisme (sans le système d'émission des signaux) à l'aide de la représentation employée pour les *schemas* de R.C. Arkin :



- le module  $I$  représente la *sélection du signal de plus forte intensité* (formule 6.2), donc le choix entre un appel ou une répulsion (fuir).
- le module  $A$  est le *test d'altruisme* (formule 6.1) permettant de définir le *but* courant de l'agent : suivre une objectif personnel ou répondre au signal extérieur précédemment sélectionné.
- le module  $\Sigma$  représente l'opération de sommation des vecteurs : évitement des obstacles, la direction du but et l'évitement des agents émettant des signaux répulsifs.

Notons qu'il est possible d'ajouter à ce modèle d'autres *motor schemas* s'ils peuvent se combiner aux premiers, par exemple un vecteur aléatoire.

Résumons les différences de notre proposition avec le modèle des *schemas* de R. C. Arkin [Arkin, 1998] :

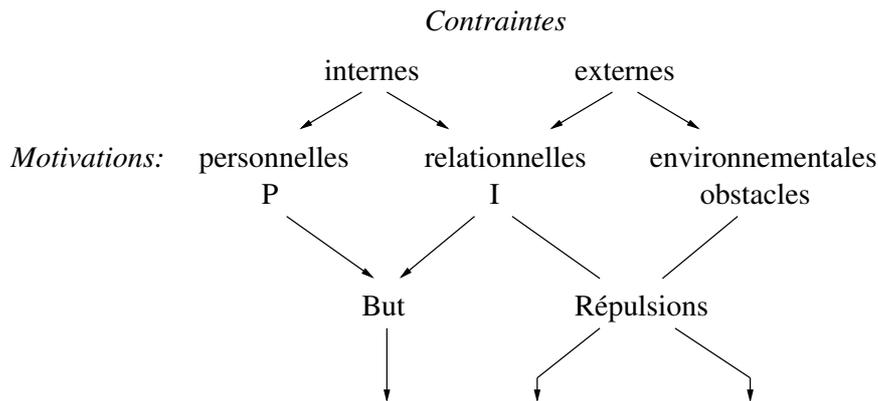
- emploi *d'un seul schema* pour définir le comportement spatial de l'agent dans diverses tâches,
- intégration des actions coopératives dans la combinaison vectorielle *et en réponse à des signaux*.
- évitement anticipé des agents par perception des signaux répulsifs (plutôt qu'une perception "visuelle"),
- choix de la tâche à mener en fonction de la satisfaction et de l'écoute en continu des signaux extérieurs,
- intégration d'un comportement altruiste - temps réel - au coeur de l'architecture.

### Combinaison vectorielle

Quels sont les vecteurs mis en jeux ?

- Le but est défini par un unique vecteur  $\vec{V}_{goal}$  calculé suivant un attracteur ou un répulseur. S'il s'agit d'une tâche pré-définie, le vecteur est proportionnel à  $\vec{V}_{agent-but}$ . S'il s'agit d'un signal, le but est calculé comme sa réaction altruiste  $\vec{\vartheta}(t)$  (formule 5.1).
- Les obstacles détectés sont contournés par le calcul d'un vecteur glissement/répulsion résultants, que nous noterons  $\vec{V}_{sli}$  (cf. principe figure 6.2).
- Les réactions aux signaux répulsifs perçus, permettant à l'agent d'optimiser sa trajectoire pour minimiser les insatisfactions de ses voisins. Chaque signal  $j$  est transformé en son vecteur altruisme de contournement noté  $\vec{V}_{alt_j}$  égal à  $\vec{\vartheta}'_j(t)$  (formule 5.3).

La formule générale de la trajectoire s'écrit :



$$\vec{V} = g_1 \cdot \vec{V}_{goal} + g_2 \cdot \vec{V}_{sli} + g_3 \cdot \frac{1}{M} \sum_{j=1}^M k_j \cdot \vec{V}_{alt_j} \quad (6.3)$$

Les coefficients  $k_j$  définissent le poids de chaque vecteur altruisme, tels qu'ils soient proportionnels à l'intensité des signaux.

### Cohérence du comportement et paramètres de la somme :

K. Zeghal a montré dans [Zeghal, 1994] qu'un agent pouvait - sous certaines conditions - se déplacer sans collisions vers un but situé dans un environnement contenant des obstacles

fixes ou mobiles par application de la formule générique suivante :

$$\vec{A}_i = \vec{A}_{goal_i} + \sum_{j \neq i}^N \beta_{ij} \cdot \vec{A}_{ij} \quad (6.4)$$

où  $\vec{A}_{goal_i}$  est l'attraction de l'agent  $A_i$  face au but et  $\vec{A}_{ij}$  représente l'action de  $A_i$  face à l'obstacle  $B_j$  tel que :

- l'action face à un obstacle "éloigné" est le calcul d'un vecteur glissant tangent à l'équipotentielle du champ répulsif formé par l'obstacle. Son sens est déterminé par la projection du vecteur  $\vec{A}_i$  sur la tangente (cf. fig. 6.2.a),
- l'action face à un obstacle "très proche" est un vecteur répulsif inhibant les vecteurs glissants et le vecteur but,
- plusieurs vecteurs glissants se combinent par ordre décroissant de leur norme (cf. détails du calcul [Zeghal, 1994] p. 38-39).

Dans le cas de notre formule, nous ne considérons la présence que d'un seul obstacle (barycentre des *perceptions avant* de l'agent, cf. détails section 12.2.2). Le vecteur réaction  $\vec{V}_{sli}$  est calculé en fonction de la distance à l'obstacle tel qu'il varie progressivement d'un léger contournement (obstacle très éloigné) jusqu'à la répulsion (obstacle très proche).

Les signaux répulsifs sont transformés en vecteurs contournements  $\vec{V}_{alt_j}$  qui sont sommés suivant leur intensité. Le vecteur résultant est combiné, par un *faible poids*, aux vecteurs  $\vec{V}_{goal}$  et  $\vec{V}_{sli}$ .

Le contrôle sur les actions est défini par les poids  $g_1$ ,  $g_2$  et  $g_3$  dont les valeurs varient progressivement en fonction de la distance à un éventuel obstacle. Le tableau suivant résume le fonctionnement de ce contrôle :

pas d'obstacle	obstacle éloigné	obstacle très près
$g_2 = 0$ et $g_1 > g_3$	$g_1 > g_2 > g_3$	$g_2 \gg g_1 \gg g_3$

Ainsi, notre formule de navigation 6.3 est une écriture de la formule générique 6.4 qui vérifie ses conditions d'applications et par conséquent assure sa cohérence.

La figure 6.2.b donne un exemple de plusieurs influences perçues par un agent (A) : contournement d'un obstacle, attraction vers son but courant, répulsion d'un agent (B) et appel d'un autre agent (C).

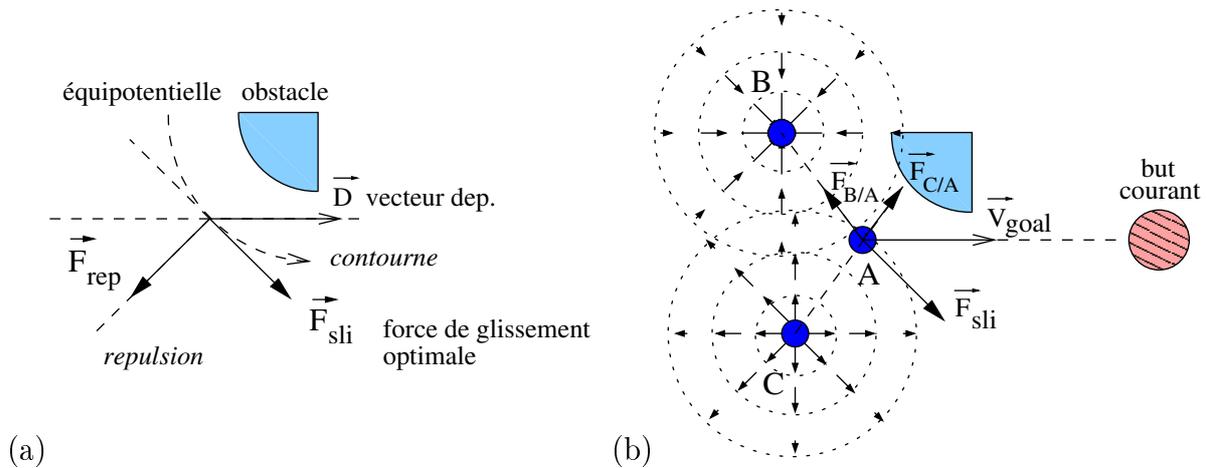
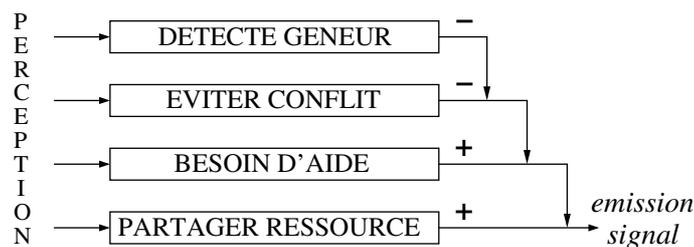


FIG. 6.2 – (a) modèle de glissement K. Zeghal [Zeghal, 1994] (b) exemples d'influences perçues par un agent dans le modèle satisfaction - altruisme

#### 6.4.4 Choix du signal à émettre

Un agent émet un signal lorsqu'au moins une des situations  $Sit_1$ ,  $Sit_2$ ,  $Sit_3$  ou  $Sit_4$  déterminées en section 5.1.1 est perçue. S'il se trouve dans une situation où plusieurs conditions d'émissions sont vérifiées simultanément, il doit faire un choix. En effet, un seul signal peut être émis à la fois puisque l'information est minimale et ne peut prendre que la sémantique d'attraction ou de répulsion.

Ce choix est défini par l'ordre de priorité donné aux comportements coopératifs et individuels en section 6.2. Nous représentons cette hiérarchie par le principe de subsomption ([Brooks, 1986]). Le signe des valeurs émises est donné à la sortie de chaque module :



La détection d'un gêneur (blocage spatial, conflit) paralysant l'activité d'un agent est la situation prioritaire pour émettre un signal répulsif (cf. chapitre 9). Autrement, la perception d'une situation pouvant engendrer un conflit (imminent) déclenche par précaution l'émission d'une répulsion (par exemple, lors d'un déplacement en milieu dense en agents).

Vient ensuite la situation critique d'un agent isolé qui a besoin d'aide pour agir ou être dépanné. Dans ce cas, il émet un signal attractif. Enfin, si toutes ces conditions ne sont pas

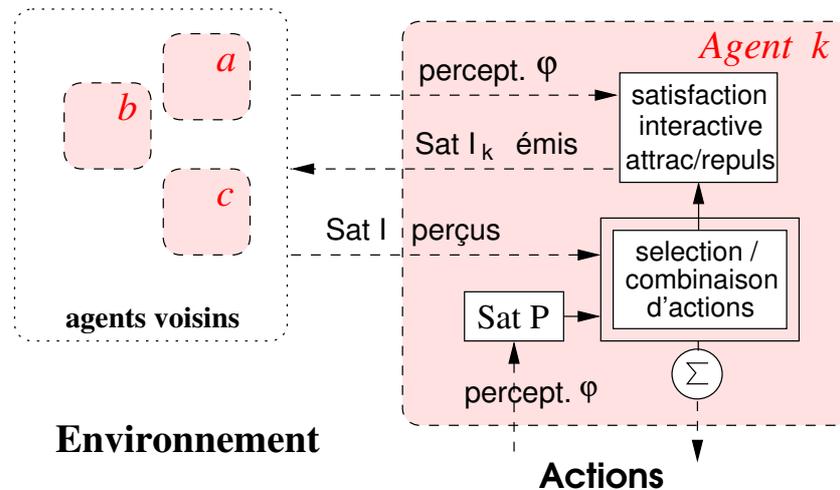


FIG. 6.3 – Le modèle Satisfaction - Altruisme

vérifiées, l'agent peut utiliser un signal attractif pour recruter ou partager une ressource abondante.

#### 6.4.5 Le modèle général

Le modèle général est représenté par le schéma de la figure 6.3. D'une part les agents s'échangent des signaux de satisfaction interactive pour s'influencer, d'autre part chacun calcule sa satisfaction personnelle suivant la progression de ses actions. Le module de sélection/combinaison d'actions applique les formules de sélections (des signaux et de l'altruisme, formules 6.1 et 6.2) et de combinaison vectorielle (formule 6.3). Le module satisfaction interactive / appels détermine si l'une des situations  $Sit_1$ ,  $Sit_2$ ,  $Sit_3$  ou  $Sit_4$  est vérifiée et peut déclencher l'émission d'un signal.

Le chapitre suivant présente le fonctionnement détaillé de l'architecture et de l'écriture de ses algorithmes. Ainsi nous pouvons évaluer sa complexité en temps et la complexité pour son implémentation en robotique.



# Chapitre 7

## Architecture et Algorithmes

### 7.1 Sélection d'une fonction

Nous venons de décrire, dans le chapitre précédent, le système d'interaction permettant aux agents de réagir en temps réel à la perception de signaux attractifs ou répulsifs. Ces signaux sont émis dans le but de réduire les gênes spatiales et d'assurer des comportements coopératifs. Pour cela, chaque agent compare son niveau de satisfaction personnelle avec les signaux perçus pour décider d'un comportement altruiste ou au contraire de poursuivre sa tâche courante.

Les fonctions (ou sous-tâches) des agents sont des comportements pré-définis qui sont déclenchés suivant des perceptions de stimuli. Cette section précise la définition de ces comportements ainsi que le processus de sélection de la fonction courante exécutée par l'agent.

#### **Fonctions (ou comportement)**

Nous avons présenté en introduction du chapitre 4 qu'une action ou une fonction de l'agent se déclenchait si certaines conditions internes (motivations) et externes (stimuli, perception d'une situation nécessitant une action) étaient vérifiées. C'est une démarche classique pour des architectures à base de comportements compétitifs (ex. [Drogoul, 1993]).

Nous présentons notre implémentation de ce principe au travers du modèle des satisfactions et des signaux d'influences.

Le comportement d'un agent consiste à exécuter continuellement l'une des fonctions dont il dispose, nommée fonction ou tâche courante. Le processus de *contrôle* ou de *sélection*

*d'actions* fournit à l'agent un moyen de sélectionner la fonction la mieux adaptée à sa situation courante. Le modèle développé dans cette thèse permet de tenir compte de trois type de motivations :

- la motivation à exécuter une tâche en fonction de son importance et de la perception de stimuli déclencheurs,
- la motivation à poursuivre la tâche en cours,
- la motivation à répondre aux demandes extérieures.

Les deux premières motivations concernent les fonctions propres à l'agent, la dernière est d'ordre coopératif. Nous avons montré comment le test d'altruisme (eq. 6.1) permettait d'effectuer une sélection entre tâche individuelle et action altruiste. Toutefois, la tâche individuelle éventuellement exécutée doit elle-même être *sélectionnée* parmi les différentes fonctions dont dispose l'agent. C'est ce processus que nous développons ci-dessous.

La fonction individuelle choisie par l'agent doit répondre aux mieux à la situation dans laquelle il se trouve (celle qu'il perçoit). Cette sélection est réalisée par comparaison des perceptions courantes de l'agent avec les conditions de déclenchement de ses fonctions. Cependant, l'agent doit posséder aussi une certaine motivation à *persister* dans une tâche qu'il a débuté.

Ces notions de compétitions et de persistance sont modélisées par les concepts de motivations et de satisfaction personnelle. Ainsi, nous définissons les fonctions des agents comme suit :

Chaque fonction  $f_i$  est définie par

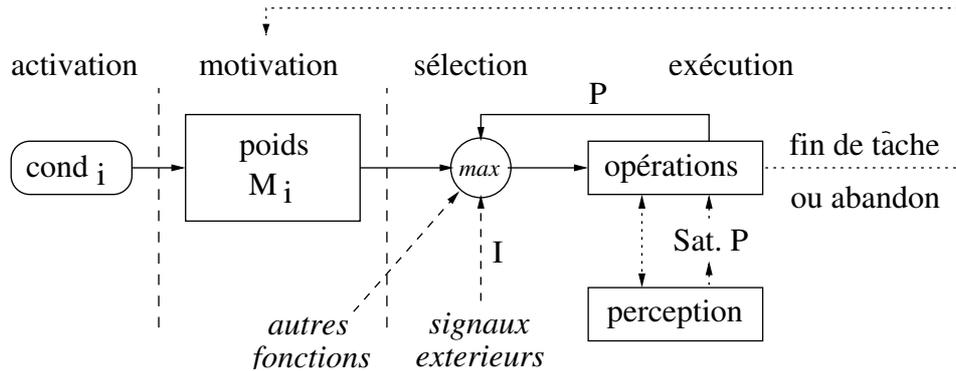
- des *conditions de déclenchements* (perception de stimuli) :  $cond_i(s) \rightarrow booleen$ ,
- un *poids* (ou importance de la tâche) :  $M_i \rightarrow [0, 1]$ ,
- des *opérations* (ordres vers les effecteurs),
- une *mesure de la satisfaction personnelle*  $P_i \in [-1, 1]$ , ( $P_{max} = 1$ ).

Les informations en provenance des capteurs ou senseurs sont notées  $s$ .

### Conditions de déclenchement

Pour être déclenchable une fonction doit tout d'abord vérifier la présence des stimuli déclencheurs (externes et/ou internes). Si ces tests sont vérifiés la fonction est dite activable :

*Si  $cond_i(s) = vrai$  alors  $f_i$  activable*

FIG. 7.1 – Décomposition d'une fonction  $f_i$  dans le modèle satisfaction-altruisme

### Motivation et satisfaction

Chaque tâche reçoit un poids d'activation initial  $M_i$ , qui représente l'importance de la sous-tâche éventuellement traitée. C'est-à-dire un niveau de satisfaction "idéal" de la fonction.

Ainsi  $M_i$  est définie comme *la borne supérieure de la satisfaction personnelle* de la fonction  $f_i$ . Le calcul de la satisfaction personnelle (donné au chapitre 4) est complété de l'opération suivante :

$$\text{Si } P(t) > M_i \text{ alors } P(t) = M_i$$

Cette borne supérieure permet d'effectuer une comparaison cohérente entre la valeur de satisfaction personnelle courante et le poids de déclenchement de la fonction (cf. sélection de tâche ci-après).

Enfin, précisons que le poids d'une tâche peut varier suivant la valeur de satisfaction personnelle obtenue lors de son exécution. Nous notons ainsi la valeur du poids de déclenchement courant d'une fonction  $M_i(t)$ , avec  $M_i(0) = M_i$ . Nous développons plus loin le processus de variation de ces poids.

### Processus de Sélection

Dans notre modèle, les fonctions activables sont continuellement en compétition, afin de déterminer en temps réel la plus pertinente à exécuter. Nous distinguons trois types de fonctions : la fonction courante activée, les fonctions individuelles non activées et la réaction altruiste à des signaux extérieurs (cf. fig. 7.1).

Notons bien que lorsqu'une fonction est déclenchée, elle continue à être en compétition avec les fonctions potentiellement activables et les signaux perçus. Lors de la phase d'exécution, la satisfaction personnelle joue un rôle essentiel dans l'évaluation de la force de continuation de la fonction active.

Le processus de compétition calculé à chaque cycle de délibération de l'agent se décompose comme suit :

Dans un premier temps, il faut déterminer, parmi les fonction activables, celle de plus grande motivation :

Soit  $F$  l'ensemble des fonctions activables :  $F = \{\forall i f_i / \text{cond}_i(s) = \text{vrai}\}$ ,

Le processus de compétition consiste à conserver la fonction activable de motivation maximale. Soit  $F_m$  l'ensemble des fonctions de plus grandes motivations de  $F$  :

$$f_k \in F_m : \forall f_i \in F \quad M_k(t) \geq M_i(t)$$

Si  $F_m$  contient plusieurs fonctions, le choix de la nouvelle fonction potentiellement activable est déterminé par un tirage aléatoire<sup>1</sup> :

$$f_m = \text{alea}(F_m)$$

Le test d'altruisme met en compétition une tâche individuelle et des signaux extérieurs. Cette tâche individuelle peut être soit la fonction courante de l'agent  $f_c$ , soit la nouvelle fonction potentiellement activable  $f_m$ .

Le choix entre une nouvelle fonction  $f_m$  et la fonction courante se fait par comparaison de la motivation  $M_m(t)$  avec la satisfaction personnelle  $P(t)$  de la fonction courante. Toutefois, pour permettre à l'agent de persister dans l'exécution de sa fonction courante (rechercher une solution en cas de difficultés), nous diminuons d'une constante  $\gamma \in [0, 1]$  la motivation de la fonction  $f_m$ . Le test de comparaison s'écrit :

$$\text{Si } P(t) > M_m(t) - \gamma \text{ alors continuer } f_c \text{ sinon } f_c = f_m$$

Ce processus de sélection est illustré par le schéma figure 7.2.a.

La dernière étape de sélection consiste à effectuer le *test d'altruisme*, qui est la comparaison de la satisfaction personnelle  $P$  avec le signal extérieur le plus intense. Cependant, si une nouvelle fonction individuelle vient d'être sélectionnée, cette comparaison ne peut

---

<sup>1</sup>la fonction *alea* est définie en section 6.4.1 formule 6.2

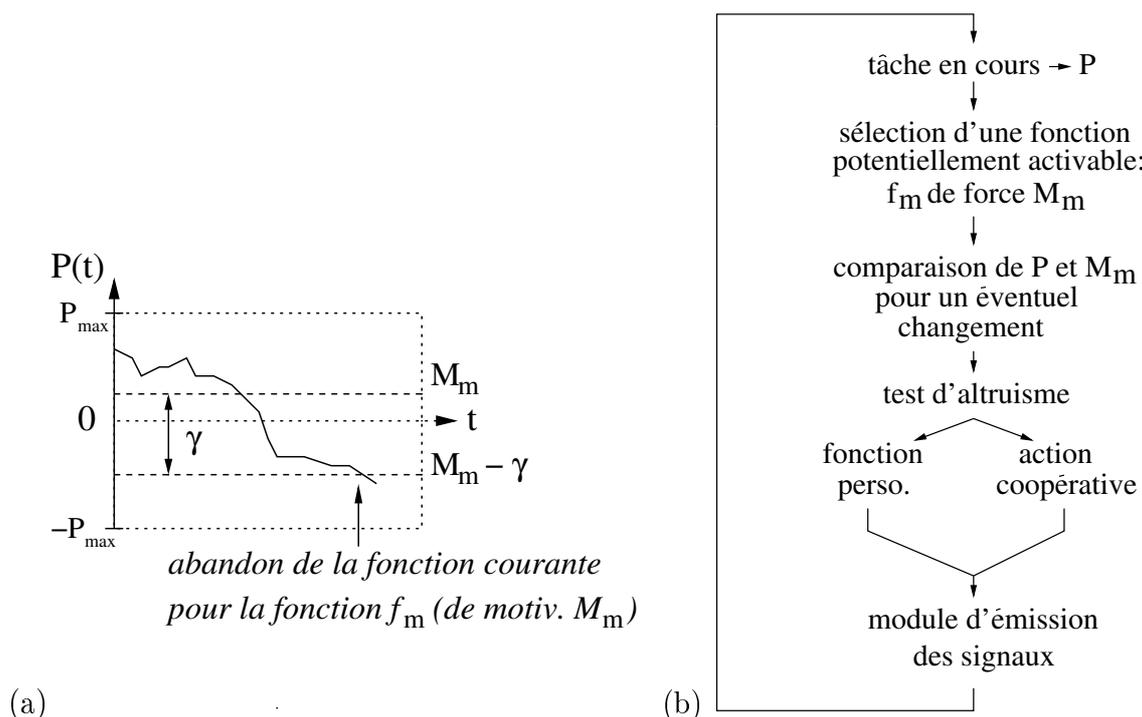


FIG. 7.2 – (a) Compétition entre la fonction courante (de motivation  $P(t)$ ) et une fonction de motivation  $M_m - \gamma$ . (b) Description du cycle perception-(ré)action du modèle.

être effectuée puisque  $P$  n'est pas encore évaluée. Dans ce cas, la satisfaction personnelle est initialisée au poids de la fonction :  $P(t) = M_m(t)$ . Ceci permet à l'agent de débiter sa tâche avec un niveau de satisfaction élevé, lui permettant de supporter temporairement des régressions dans l'accomplissement de ses actions.

Ainsi, le test d'altruisme est toujours applicable :

*Si*  $\alpha \cdot |I_{ext}| > (1 - \alpha) \cdot P(t)$  *alors* arrêter fonction en cours et prendre pour but la direction du vecteur altruisme.

La figure 7.2.b reprend l'ensemble des opérations qui viennent d'être présentées, exprimant la totalité du cycle perception-(ré)action du modèle. Celui-ci est écrit formellement sous forme d'un algorithme dans la section suivante.

### Variation des poids pour éviter les oscillations

Le modèle proposé sélectionne en temps réel une fonction en comparant les motivations des fonctions activables. Ce type de démarche "réactive" s'expose alors au risque d'oscilla-

tion entre différentes fonctions déclenchables. Ce problème peut apparaître lorsque l'agent doit choisir parmi plusieurs fonctions activables qui se révèlent difficiles ou impossibles à accomplir lors de leur exécution.

En effet, nous avons vu que lorsque une tâche est déclenchée, la satisfaction personnelle et le paramètre  $\gamma$  déterminent la persistance de la fonction exécutée. Supposons maintenant que l'agent soit en situation de choisir entre deux fonctions activables, de motivations proches, mais dont les exécutions sont très insatisfaisantes. L'agent serait alors conduit à choisir l'une des fonctions, puis à l'abandonner car insatisfaisante et à choisir aussitôt la seconde. Or, ce même schéma se reproduirait avec cette deuxième fonction, qui insatisfaisante, provoquerait à son tour le choix de la première, etc.

Ce type de cycle peut être rompu en "inhibant" toute fonction en échec. Ceci est fait dans notre modèle *en introduisant la satisfaction dans l'évaluation de la motivation* à déclencher une tâche. L'idée générale est de réduire *temporairement* le poids de déclenchement d'une fonction lorsque celle-ci délivre un niveau de satisfaction négatif. Ce processus est introduit à l'arrêt d'une fonction, par une réduction éventuelle de sa motivation :

Lorsqu'une fonction  $f_c$  est abandonnée ou terminée,  
Si  $P(t) < 0$  Alors  $M_c(t) = 0$  ,  $[M_j(t) = 0]$

Si l'état d'exécution d'une fonction est négatif ( $P(t) < 0$ ), au moment où elle se termine, son poids prend la valeur minimale 0. Optionnellement, d'autres fonctions  $f_j$  peuvent subir la même opération, ce sont des fonctions dépendant de l'exécution de  $f_c$ . Un exemple concret est donné dans le chapitre suivant section 8.3.2.

Les fonctions dont la motivation de déclenchement est temporairement "inhibée" doivent retrouver progressivement leur poids initial ; ceci dans le but de leur redonner, au bout d'un certain temps, l'opportunité d'être déclenchée. Cette opération se fait progressivement par le processus suivant :

A chaque itération du cycle :  $\forall i$  Si  $M_i(t) < M_i$  alors  $M_i(t) = M_i(t - 1) + \varepsilon$

$\varepsilon$  est une valeur petite devant 1 (elle détermine la durée d'inhibition relative). Ce processus de réduction temporaire des motivations permet d'éviter les oscillations entre différentes fonctions et ce de façon adapté à la situation de l'agent :

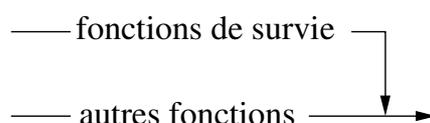
- la satisfaction personnelle permet de déclencher "l'inhibition temporaire" de toute fonction se révélant impossible ou difficile à accomplir,
- rapidement, l'agent n'a plus qu'à choisir parmi des tâches dont la motivation représente un potentiel d'accomplissement élevé (ou de la tâche par défaut, par exemple d'exploration aléatoire pour des robots mobiles).

Cette variation des poids n'est pas un processus de spécialisation de l'agent (il n'y a pas de renforcement continu du poids) mais une adaptation de l'agent à sa situation courante.

### Fonctions de survie

Nous venons de préciser que la fonction sélectionnée par l'agent était celle de motivation maximale. Si nous définissons des “*fonctions de survies*” pour gérer des situations critiques (besoin d'énergie, pannes d'un robot) nous souhaitons que celles-ci soient prioritaires. Dans ce cas, nous leur appliquons une motivation maximale ( $M_i = 1$ ). Mais il se peut que malgré la perception des stimuli déclencheurs une fonction de survie ne soit pas sélectionnée. Ce risque existe lorsque le coefficient  $\alpha$  est supérieur à 0.5 ou que  $\gamma$  est supérieur à 0.

Pour éviter ce problème, nous avons défini une classe “fonctions de survie” qui est prioritaire sur les autres. Si l'une de ces fonctions prioritaires est activable, alors elle inhibe la sélection des fonctions classiques. Ce principe est représenté par le schéma de subsomption suivant :



S'il existe plusieurs fonctions de survies l'utilisateur doit les hiérarchiser suivant leur importance pour l'agent. En particulier, nous montrerons au chapitre 9 que la réaction altruiste aux signaux répulsifs doit être considérée comme la fonction de survie la plus prioritaire, afin d'assurer la survie de chaque agent et du système (cf. [Simonin et al., 2000]).

## 7.2 Le noyau du modèle (sélection et combinaison)

### 7.2.1 Algorithme

L'algorithme suivant décrit le cycle générique du modèle satisfaction-altruisme. Toutes les opérations sont détaillées à l'exception du module de réception et d'émission des signaux que nous développons plus loin.

A l'entrée du cycle, l'algorithme connaît la fonction courante active  $f_c$  (l'état d'altruisme est codé comme une fonction de déplacement), l'ensemble des signaux perçus  $I_1..I_n$  et toutes les informations en provenance des capteurs proprioceptifs et extéroceptifs (*Sensors*).

**Algorithm 1:** Algorithme générique d'un cycle du modèle satisfaction-altruisme

---

**Data** :  $f_c, I_1, I_2 \dots I_n, \text{Sensors}$

- 1 Evaluer  $P(t)$  pour  $f_c$ , limiter sa valeur à  $M_c$   
 Pour tout  $M_i(t) < M_i$  faire  $M_i(t) = M_i(t) + \epsilon$
- 2 **si** fonction de survie activable **alors**
  - | **si** déplacement **alors** calculer  $\vec{V}_{goal}$
  - fin**
  - sinon**
- 3 |  $F \leftarrow \{\forall i f_i \text{ tel que } \text{cond}_i(\text{Sensors}) = \text{vrai}\}$   
 |  $f_m \leftarrow \text{random}(f_k \in F : \forall i M_k(\text{Sensors}) \geq M_i(\text{Sensors}))$   
 | **si**  $P(t) < M_m(t) - \gamma$  **alors**  
 | | **si**  $P(t) < 0$  **alors**  $M_c(t) = 0$   
 | |  $f_c = f_m ; P = M_m$   
 | **fin**
- 4 |  $I_{max} = \text{random}(\text{prio-int}(I_1, I_2, \dots, I_n))$   
 | **si**  $\alpha \cdot |I_{max}| > (1 - \alpha) \cdot P$  **alors**  $f_c = \text{altruisme}; \vec{V}_{goal} = \vec{\mathcal{J}}_{I_{max}}$   
 | **sinon**  
 | | **si**  $f_c$  est un déplacement **alors**  $\vec{V}_{goal} = \vec{V}_{f_c}$   
 | **fin**
- fin**
- si** fonction de déplacement **alors**
- 5 | Intégrer les signaux répulsifs :  $R = \{\cup I_i / I_i < 0\}$  ;  $\vec{V}_{rep} = \sum_{r \in R} k_r \vec{\mathcal{J}}_r$
- 6 | Intégrer le contournement des obstacles  $O$  :  $\vec{V}_{sli} = \sum_{j \in O} \vec{V}_{sli_j}$
- fin**
- si** action = déplacement **alors** Calculer  $\vec{V} = g_1 \cdot \vec{V}_{goal} + g_2 \cdot \vec{V}_{obs} + g_3 \cdot \vec{V}_{rep}$
- Transmettre l'action aux effecteurs
- Executer module de reception et d'emission des signaux

---

### 7.2.2 Un processus temps réel

L'objectif de ce paragraphe est de vérifier que le modèle satisfaction-altruisme est capable d'assurer à l'agent des prises de décisions en temps réel. Pour cela, nous évaluons la complexité en temps de l'algorithme. Mais, comme celui-ci utilise des informations provenant des capteurs nous envisageons sa complexité suivant divers types d'implémentations.

Définitions d'un système temps réel :

- "A real-time computer system may be defined as one which controls an environment by receiving data, processing them, and returning the results sufficiently quickly to affect the environment at that time." [Yourdon, 1989]

Les opérations sont évaluées suivant une mesure classique de la complexité en temps, nous précisons un ordre de grandeur concret chaque fois que cela est possible. Seules les étapes (lignes) qui exécutent plus qu'une simple instruction sont commentées :

- ligne 1 : Nous avons présenté au chapitre 4 le calcul de la satisfaction personnelle à partir de la progression, à chaque pas du cycle, de la tâche de l'agent. Cette mesure incrémentale est réalisée à partir des différents capteurs de l'agent. Pour conserver un aspect temps réel, cette évaluation ne doit pas être trop complexe, c'est-à-dire faisant appel à des capteurs de haut niveaux (ex. caméra). Les chapitres suivants donnent des expressions simples de cette mesure (en simulation et pour des robots réels).

- ligne 2 : Les tests déclenchant les fonctions de survie sont essentiellement liés à des informations provenant des capteurs internes de l'agent. Ces informations sont accessibles facilement et immédiatement (cablées).

- ligne 3 :  $O(m)$  ( $m$  est le nombre de fonctions de l'agent). Dans la pratique, le nombre de fonctions d'un agent est relativement petit ( $<100$ ), ainsi le temps de calcul pour rechercher la tâche ayant le niveau d'intensité le plus élevé est *très court*. Toutefois, ce calcul s'accompagne d'une vérification de la validité des conditions de déclenchement de chaque tâche. Ces tests sont codés de manière à être *non bloquants et immédiats*  $O(1)$ , c'est-à-dire que l'algorithme accède une mémoire contenant le résultat de processus parallèles chargés de les réaliser. Sans cette hypothèse, l'exécution de certains tests pourrait ralentir le cycle comportemental (par exemple l'analyse d'images en provenance d'une caméra embarquée). Notons que la *réactivité de l'agent* aux événements reste dépendante de la durée réelle de ces tests. Lors de nos implémentations sur des mini-robots (chapitre 12) nous n'avons employé que des tests très simples sur des perceptions en provenances de capteurs infra-rouge et de variables internes (assurant un fonctionnement temps réel).

- ligne 4 :  $O(n)$  ( $n$  est le nombre de signaux perceptibles par un agent). Les informations perçues sont stockées dans un vecteur de taille  $n$ , toujours accessible par l'algorithme. Ainsi, le calcul du signal de plus forte intensité est réalisé par une simple

boucle de  $n$  tests. Dans la pratique, le nombre de signaux perçus par un agent à un instant donné est très petit (à cause de la localité des signaux émis), c'est-à-dire très inférieur à 100.

- ligne 5 :  $O(n)$  Dans le pire des cas, tous les signaux perçus peuvent être répulsifs. Ainsi, le calcul du vecteur altruisme répondant à tous ces signaux est une somme de  $n$  vecteurs altruismes. Le calcul de la formule du vecteur altruisme nécessite un nombre d'instructions mathématiques faible. L'obtention des informations  $\overrightarrow{BA}$  et  $I$  est peu coûteuses en temps car elle nécessite une simple évaluation de la réception du signal. En effet, le signal contient directement la valeur  $I$  et plusieurs récepteurs peuvent permettre de déduire la direction de l'émetteur (cf. chapitre 12). Ainsi, le temps de calcul d'un vecteur altruisme est borné par une petite constante.

- ligne 6 : Il s'agit de transformer la détection des obstacles (face à l'agent) en un vecteur répulsion. La complexité d'une telle opération dépend du moyen utilisé pour réaliser la perception de l'environnement. Généralement, les robots mobiles sont munis de perceptions temps réel tels que : ondes infrarouge (ex. robot Khepera [de Meneses and Carmona, 1999]), ultra-sons (ex. ZVD [Zapata et al., 1995]), caméra à vision linéaire (ex. 64 pixels pour des Kheperas). Avec de tels systèmes de perceptions, on peut estimer en temps réel un vecteur moyen représentant le barycentre des obstacles entourant un robot.

Finalement, l'algorithme exécute *séquentiellement* des instructions simples et des boucles linéaires (lignes 3, 4, 5 et 6). Nous pouvons affirmer que l'agent implémente un modèle temps réel si les perceptions des stimuli se font aussi "en temps réel". C'est-à-dire suffisamment rapidement devant "l'urgence" des situations (donc en particulier suivant le temps de réaction de l'agent devant les obstacles fixes et mobiles). Dans ces conditions, le temps de réaction n'est limité que par l'inertie des mouvements des mécanismes.

Le chapitre 12 présente le développement de robots mobiles capables de remplir ces objectifs jusqu'à une certaine vitesse limite.

### 7.3 De l'interaction à la propagation d'informations

Jusqu'à présent nous n'avons considéré les signaux entre agents que comme des influences directes et locales. Cependant, le système étant composé de plusieurs agents en interactions, les effets d'une action peuvent se propager au delà de leur conséquences initiales.

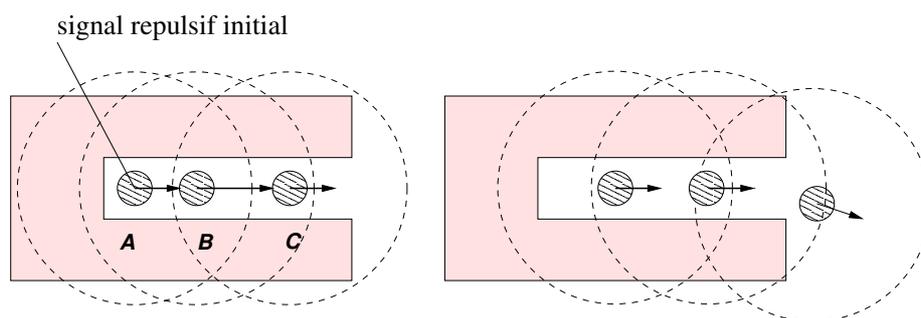


FIG. 7.3 – Propagation implicite des répulsions

### Propagation implicite des répulsions

Examinons la situation présentée en figure 7.3 : il s'agit de trois agents réalisant des tâches indépendantes dans un étroit passage. Supposons que l'agent *A* décide de sortir de ce passage, il se trouve alors gêné par son voisin *B*. Pour tenter de le repousser il émet en continu un signal répulsif. Ce deuxième agent perçoit la demande et devient altruiste, mais il ne peut y répondre car il est lui-même bloqué par la présence de troisième agent *C*. A son tour *B* tente de repousser ce gêneur en émettant un nouveau signal répulsif. Finalement, le troisième agent *C*, qui peut se déplacer, libère les deux premiers.

Cette chaîne répulsive a débuté par l'émission d'un signal qui exprimait seulement l'insatisfaction d'un agent envers un autre. Or cette insatisfaction s'est propagée chaque fois qu'elle n'a pas pu être résolue localement. Cette propagation des contraintes est donc émergente puisqu'elle n'est pas prévue initialement dans le modèle.

Cet aspect fondamental du modèle a été développé et affiné pour traiter les problèmes de conflits spatiaux entre robots mobiles autonomes. Ce travail est présenté en détail au chapitre 9.

### Propagation explicite des attractions

Un tel type de propagation existe-t-il lors des émissions d'attractions ? A l'évidence non, puisque si un agent ne peut rejoindre celui qui l'appelle c'est qu'il est gêné par un voisin ou un obstacle. Dans ce cas, c'est un signal de répulsion qu'il pourra éventuellement émettre, et non une nouvelle attraction.

Nous nous sommes tout de même intéressé à la possibilité de créer volontairement des

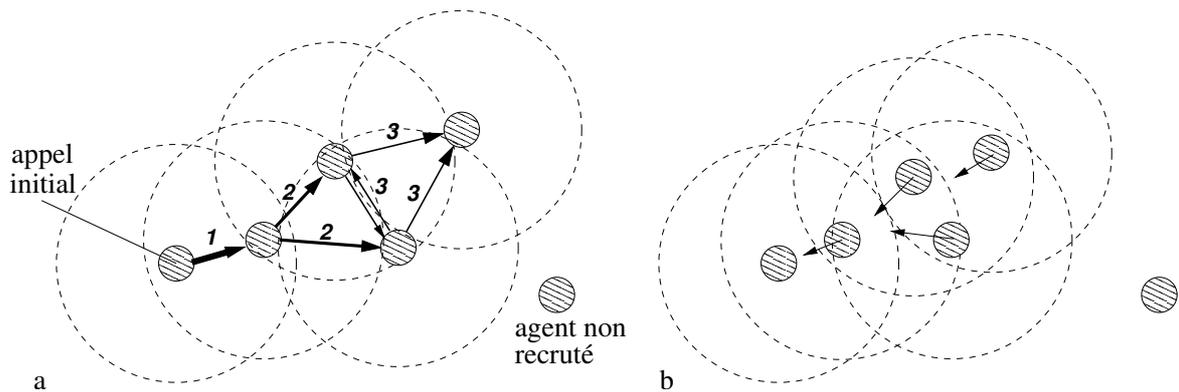


FIG. 7.4 – Principe de propagation explicite des attractions

propagations d'attractions entre les agents. En effet, il apparaît que dans de nombreuses situations un signal d'appel ne s'adresse pas qu'à un seul agent mais à plusieurs. C'est le cas pour le recrutement et le partage d'une ressource abondante.

Ainsi, par effet de propagation des attractions entre agents, il est possible de considérablement accélérer une opération de recrutement ou de partage. La figure 7.4 présente le principe de propagation des attractions.

Supposons qu'un agent ait trouvé un objet devant être transporté, mais qu'il nécessite la présence simultanée de plusieurs agents. Si cet agent émet un signal d'appel dans une zone peu peuplée, il risque d'attendre longtemps l'arrivée du nombre d'agents requis pour l'opération. Par contre, il suffit que le premier agent contacté émette à son tour une attraction pour agrandir la zone d'appel et donc accélérer le processus.

L'application de cette simple règle locale permet de propager très facilement à plusieurs agents une attraction locale. De plus, le modèle garanti que seuls les agents inoccupés et non contraints vont répondre à cette attraction et se diriger vers la source.

Toutefois, l'implémentation proposée de cette technique intègre un coefficient d'amortissement afin de limiter l'étendue potentielle de la propagation. En effet, il ne faut pas qu'un seul signal attractif puisse attirer à lui tous les agents du système !

Nous avons traité ce problème par l'application de deux règles complémentaires et locales :

- Dès que le ou les agents appelant perçoivent qu'ils n'ont plus besoin de recruter, ils stoppent leur émission. Cela a pour effet immédiat de faire disparaître la chaîne d'attraction, puisqu'elle n'existait que par la propagation du signal attractif initial.
- Les agents propagent un signal avec une intensité plus faible que celui qu'il perçoivent. De plus, à partir d'un certain seuil minima, ils ne propagent plus l'attraction perçue.

La deuxième règle a pour conséquence d'atténuer l'attraction initiale au cours de la propagation et de la stopper à une certaine distance maximale (cf. fig. 7.4.a où l'intensité des attractions est représentée par des flèches plus fines).

En outre, cette *distance* ne s'exprime pas dans la métrique de l'environnement physique mais dans celle du graphe formé par les agents se propageant des signaux (cf. numérotation de propagation fig. 7.4). Cette particularité est très intéressante car elle assure un recrutement adapté à la situation géographique des agents sans que ceux-ci n'en soient conscients. En effet, si les agents sont nombreux et proches de l'appel initial, la propagation se fait rapidement et localement. Si les agents sont plus dispersés, ce même recrutement aura lieu si les agents sont "à distance de communication", la propagation agit entre les agents sans dépendre de leur distance géographique. Cette technique est évaluée dans le prochain chapitre.

### 7.3.1 Module d'émission et de propagation des signaux

Nous donnons l'algorithme générique du module de décision et de calcul des signaux à émettre. Celui-ci implémente la hiérarchie des situations prioritaires données en section 6.4.4 ainsi que le processus de propagation explicite des attractions :

---

#### Algorithm 2: Sélection du type de signal à émettre

---

```

si perception d'un gêneur ou d'un conflit imminent alors
  | émettre un signal répulsif
fin
sinon
  | si besoin d'aide ou de recruter alors
  | | émettre un signal attractif
  | fin
  | sinon
  | | si état d'altruisme et perception attraction  $I_e > Sp$  alors
  | | | émettre attraction de valeur  $I_e/k$ 
  | | | fin
  | | fin
  | fin
fin

```

---

L'algorithme assure une priorité aux situations nécessitant des signaux répulsifs (bloques, gênes, conflits imminents), puis des signaux attractifs (appel à l'aide, recrutement).

Si l'agent est en état d'altruisme et qu'il n'est pas en conflit ou en panne il peut relayer une attraction  $I_e$  pour propager un recrutement. Dans ce cas le signal perçu doit être supérieur au seuil  $Sp$  et le signal émis égal à  $\frac{I_e}{k}$  ( $k$  est une constante).

La perception des situations, ainsi que le calcul exact des valeurs à émettre sont dépendant des applications traitées par les agents. Nous donnons dans la quatrième partie du document différentes applications du modèle, en précisant pour chacune l'ensemble de ces paramètres.

Avant de développer les applications menées à partir de ces algorithmes nous donnons un bref comparatif de notre modèles avec deux architectures importantes d'agents autonomes coopératifs.

## 7.4 Comparaison à d'autres architectures

### 7.4.1 ALLIANCE (L. Parker)

Cette architecture est centrée sur le problème de la sélection d'actions d'agents *coopératifs en équipes hétérogènes*. Les comportements sont définies par des ensembles d'actions primitives et un niveau de motivation. Les primitives peuvent se combiner ou s'inhiber (suivant un schéma de subsomption) [Parker, 1994].

Le rôle de la motivation est d'activer ou non ces comportements. Celle-ci est calculée par composition d'information binaires et d'informations sur les agents voisins. Pour chaque robot, la motivation d'un comportement est calculée par la formule suivante :

$$m_{ij}(0) = 0, \text{ et}$$

$$m_{ij}(t) = [m_{ij}(t-1) + \textit{impatience}_{ij}(t)] * \textit{sensory\_feedback}_{ij}(t) * \textit{activity\_suppression}_{ij}(t) * \textit{impatience\_reset}_{ij}(t) * \textit{acquiescence}_{ij}(t)$$

$\textit{impatience}_{ij}(t)$  est un taux "d'impatience" de voir accomplie la tâche  $j$ , variable suivant que l'un des voisins informe qu'il a récemment ou non accomplie cette tâche. La formule utilise quatre prédicats *binaires* :

- $\textit{sensory\_feedback}_{ij}(t)$  indique si les pré-conditions de déclenchement du comportement sont vérifiées,
- $\textit{activity\_suppression}_{ij}(t)$  vaut 0 si l'agent exécute déjà un autre comportement que  $j$ ,
- $\textit{impatience\_reset}_{ij}(t)$  vaut 0 si l'un des agents voisins réalise déjà le comportement  $j$  (information par messages),

- $acquiescence_{ij}(t)$  représente l'abandon d'un comportement au bout d'un temps pré-fixé fonction de l'activité des agents voisins.

Ainsi, la motivation dépend de la notion d'impatience qui augmente avec le temps, mais qui peut être annulée par la réception d'informations extérieures liées aux activités des voisins. Les messages sont, comme dans notre modèle, diffusés localement et sans retour. Ils sont de nature relativement simple puisqu'ils contiennent seulement le numéro identificateur de l'agent émetteur et le numéro de sa tâche en cours.

Spécificités et critiques de l'architecture ALLIANCE :

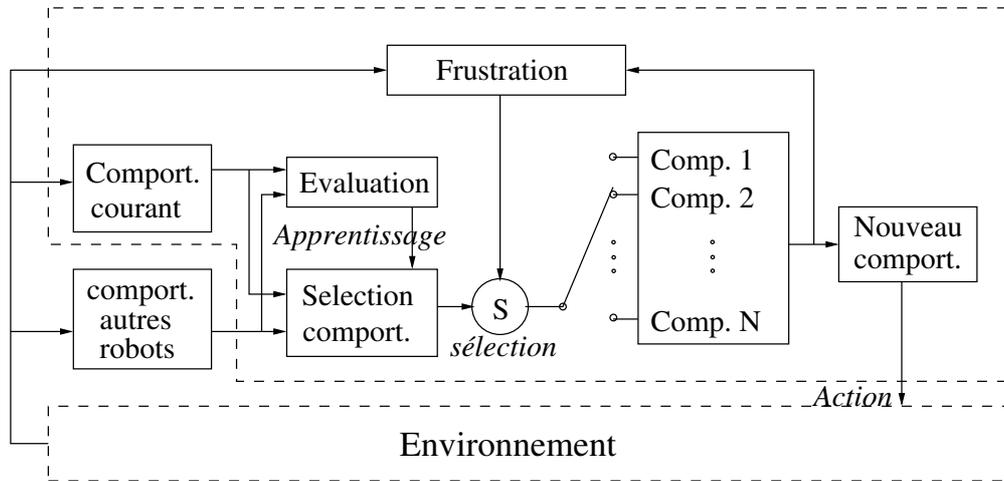
- Chaque agent doit stocker une table d'information de tous les messages reçus,
- Le traitement est limité à celui de la sélection d'actions optimisé pour la *répartition* des tâches (ou comportements),
- La nécessité d'identifier les agents et leurs comportements n'est pas adapté aux systèmes ouverts, ni composés de nombreux agents,
- L'abandon d'une tâche en cours est basée sur une simple mesure du temps écoulé,
- Une tâche est déclenchée par augmentation de l'impatience de la voir accomplie (si elle n'est pas effectuée par un autre agent).

Ces caractéristiques montrent les limites d'une approche communicante utilisant *l'identification* des agents et leurs activités, dont en particulier le besoin de stocker les informations reçues. Le modèle satisfaction-altruisme répond d'une part au problème de la sélection des comportements en évaluant les interactions compatibles-incompatibles, et d'autre part, introduit une forme de coopération intentionnelle assurant d'autres activités que la répartition dynamique des tâches. Ces deux processus sont effectuées en temps réels par l'évaluation de la satisfaction personnelle et par la diffusion de signaux de satisfaction interactive.

#### 7.4.2 Une architecture coopérative et émotionnelle

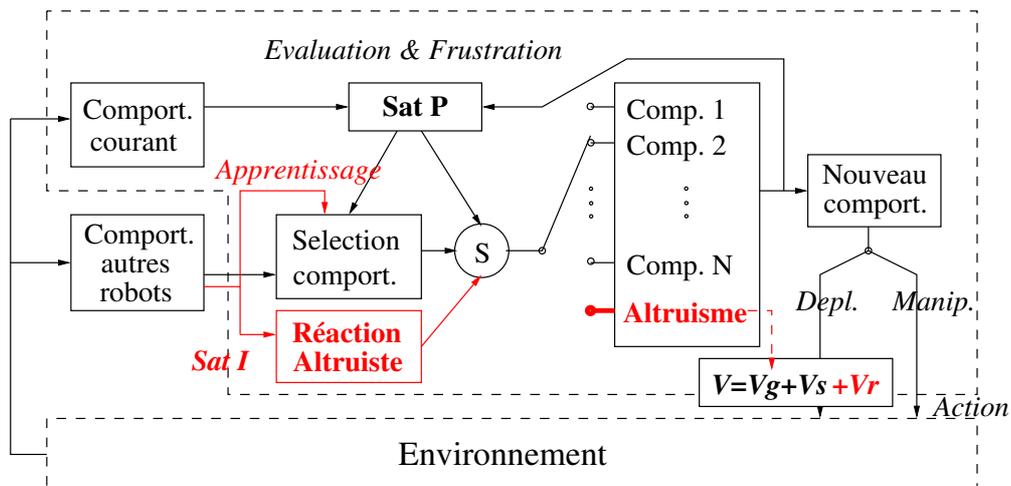
En 1996, T. Shibata, K. Ohkawa et K. Tanie ont proposé une architecture de robots mobiles coopératifs reposant sur des considérations proches des nôtres [Shibata et al., 1996] [Ohkawa et al., 1998]. Leur objectif était de concevoir des robots capables de sélectionner le meilleur comportement (ou sous-tâche) possible en fonction d'états internes et de la connaissance des consommations en énergie des agents.

Ils ont ainsi proposé le modèle de “coopération spontanée” suivant :



Cette architecture utilise des modules d'*observation* des agents voisins, d'*évaluation* et de *frustration*. Le premier représente l'échange de messages contenant la dépense d'énergie de chacun des agents pour leur tâche en cours. Le module d'évaluation (ou plus exactement de récompense) combine la mesure de l'énergie dépensée par l'agent et ses voisins ainsi qu'une récompense donnée par *un opérateur extérieur*. Enfin, le module de frustration permet de tenir compte des contraintes environnementales, c'est-à-dire qu'il évalue la réalisation de la tâche sélectionnée (définie comme émotion par les auteurs). Cette évaluation est une mesure directe du temps écoulé, au delà d'un délai de non accomplissement, la tâche est remise en cause.

Nous avons ré-exprimé le modèle satisfaction-altruisme dans la représentation utilisée par T. Shibata :



Les spécificités de notre modèle sont :

- la fusion des concepts d'évaluation individuelle et de frustration dans celui de la satisfaction personnelle (avec une mesure plus fine que le temps écoulé),
- l'échange de signaux de satisfaction interactive, c'est-à-dire d'influences, qui sont déterminés par des mesures de satisfactions (nous n'utilisons pas d'opérateur extérieurs),
- l'introduction de comportements altruistes, permettant d'intégrer des actions coopératives immédiates. Les comportements sont ainsi une combinaison/sélection des influences internes et externes.

Remarque : notre architecture peut être complétée d'un module d'apprentissage placé également à la réception des informations externes (cf. développement chapitre 10).

Notons que l'architecture de T. Shibata n'a été évaluée que par simulations, mais ses résultats confirment l'aspect adaptatif et coopératif de l'approche.

## 7.5 Organisation des chapitres suivants

Partie 4	Application à la résolution de problèmes et Simulations	Chapitre
coopération	Évaluation de l'architecture coopérative sur le problème des robots fourrageurs	8
conflits spatiaux	Traitement des conflits spatiaux par le modèle, définition des signaux répulsifs	9
hétérogénéité	Évaluation du modèle sur un système hétérogène (objets, fonctions et types de robots différents)	10
apprentissage	Intégration d'un module d'apprentissage par renforcement exploitant le modèle des satisfactions	10

Partie 5	Communication et expérimentations en robotique mobile	Chapitre
signaux	Proposition d'un protocole (couches physique et de liaison) de communication dédié aux SMA situés réels	11
robots altruistes	Intégration du modèle à des mini-robots mobiles, évaluation par des expérimentations de résolution de conflits spatiaux	12



## Quatrième partie

Application à la résolution de problèmes  
distribués et Simulations



# Chapitre 8

## Des robots fourrageurs

*The objective is to explore a distant planet, more concretely, to collect samples of a particular type of precious rock. The location of the rock samples is not known in advance, but they are typically clustered in certain spots. A number of autonomous vehicles are available that can drive around the planet collecting samples and later reenter the a mothership spacecraft to go back to earth. There is no detailed map of the planet available, although it is known that the terrain is full of obstacles-hills, valleys, etc.-which prevent the vehicles from exchanging any communication.<sup>1</sup>*

d'après Luc Steels [Steels, 1989], adapté par M. Wooldridge [Weiss, 2000]

### 8.1 Présentation du problème

#### 8.1.1 Introduction

Le problème des robots fourrageurs (aussi appelé robots explorateurs) est apparu dans la communauté intelligence artificielle vers la fin des années 80 (L. Steels [Steels, 1989], R. Brooks [Brooks et al., 1990]). Son énoncé, proposé par Luc Steels, est donné en introduction de ce chapitre car il présente de façon claire, attractive et exhaustive le problème abordé.

Toutefois, je qualifierais cette adaptation de M. Wooldridge de version “sans commu-

---

<sup>1</sup>L'objectif est d'explorer une planète distante, plus concrètement, de collecter des échantillons d'un type particulier de roche précieuse. L'emplacement des échantillons de roches n'est pas connu à l'avance, mais ils sont typiquement regroupés en certains lieux. Plusieurs véhicules autonomes sont capables de parcourir la planète en collectant des échantillons et plus tard de retourner au vaisseau mère pour les ramener sur terre. Il n'y a pas de carte détaillée de la planète, toutefois il est connu que le terrain est rempli d'obstacles-collines, de vallées, etc. - qui empêchent les véhicules d'échanger toutes communications.

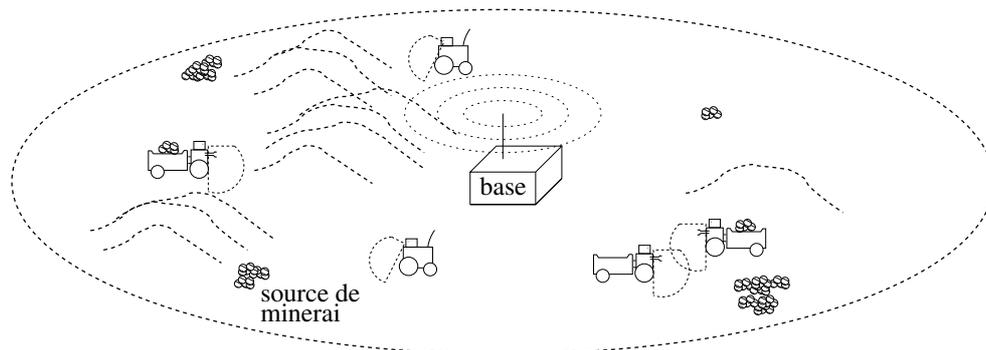


FIG. 8.1 – Représentation d'un système de robots fourrageurs

nications”, car celles-ci sont supposées impossibles ! Pourtant, il est difficile d’imaginer des terrains qui soient à la fois explorables par des robots et interdisant toutes communications, même locales. Nous supposons donc dans ce travail que des communications entre robots *voisins* sont possibles (cf. détails plus bas) et nous précisons toujours la nature de l’environnement. Le problème était d’ailleurs initialement proposé par L. Steels sans contraintes explicites sur les communications [Steels, 1989] (J. Ferber donne aussi une version générale du problème avec communications dans [Ferber, 1995]).

Les agents considérés dans ce problème sont appelés “robots fourrageurs” par analogie avec les fourmis qui effectuent des tâches de fourragements. En effet, les robots doivent *trouver*, *recupérer* et *transporter* des échantillons comme peuvent le faire des insectes avec des brindilles [Drogoul, 1993].

Au delà de cet énoncé général, il faut préciser quels sont les enjeux de l’étude d’un tel problème. J. Ferber propose dans [Ferber, 1995] cinq étapes à aborder pour le traiter et qui sont autant de sous problèmes à résoudre :

1. Définir les différents types de robots à construire (homogènes ou hétérogènes ?)
2. Déterminer les capacités cognitives des agents (type d’architecture et de communications, intégration de l’apprentissage ?)
3. Décrire les structures collectives de travail (travail en groupes ou solitaires ? équipes fixes ou dynamiques ? processus de recrutements ?)
4. Définir les mécanismes de coopération et les protocoles d’interactions (comportements en cas de conflits, actions collectives)
5. Être capable d’évaluer les choix retenus et de les comparer à d’autres organisations.

Cet ensemble de points à étudier montre que ce problème relève autant de la robotique que de l’intelligence artificielle distribuée. Ce chapitre est consacré à l’évaluation du modèle

comportemental proposé à travers le problème du fourragement. Nous précisons donc, en complément des aspects de programmation, les choix adoptés sur ces cinq points définissant la méthode de résolution.

Le premier point concerne la nature même des robots, doivent-ils être identiques ou non ?

Le problème des robots fourrageurs peut être traité par un groupe hétérogène d'agents, par exemple avec trois types de robots différents : explorateurs, transporteurs et ravitailleurs. Mais généralement il se prête bien à une résolution par des systèmes homogènes [Steels, 1989] [Drogoul, 1993]. C'est-à-dire des systèmes où chaque agent est pourvu des mêmes capacités que les autres. Ceci est assez naturel car tous les échantillons détectés ne peuvent subir qu'une seule et même opération : être chargés et transportés à la base. De plus, un tel système est très robuste, de par la redondance des fonctions chez tous les agents. L'hétérogénéité entre agents prend tout son intérêt lorsque les éléments manipulés sont eux-même hétérogènes ou lorsqu'ils sont susceptibles de subir des opérations différentes.

Les travaux que nous présentons dans ce chapitre concernent essentiellement le traitement du problème des robots fourrageurs par un ensemble *homogène* d'agents. Toutefois, pour montrer l'adéquation de notre modèle avec les systèmes hétérogènes, nous avons défini une variante du problème (nommé "traitement de déchets toxiques") que nous présentons en section 10.1.1.

### 8.1.2 Des robots footballeurs ?

Depuis une dizaine d'années, un autre type d'application fait référence dans le monde de la robotique collective, ce sont les compétitions de *robots footballeurs*. Il s'agit là aussi de plusieurs agents autonomes qui doivent coopérer pour réaliser un but commun, mais face à un adversaire qui l'en empêche (cf. [Asada et al., 1998]). Il faut toutefois noter les fortes différences qui existent entre ce problème et celui des robots fourrageurs. Les spécificités du problème des robots footballeurs sont :

- un *seul objet* est recherché et manipulé par les agents (le ballon), celui-ci est presque toujours en mouvement.
- les agents sont organisés en équipes *fixes* avec souvent des rôles initialement attribués (gardien, attaquants, défenseurs, etc.)
- les équipes sont toujours en *compétition* pour l'obtention du ballon et du gain de la partie ; mais il y a aussi coopération au sein de chaque équipe.

- Ce problème exige des agents des capacités particulières : se déplacer très rapidement, posséder une perception du jeu très précise (détection du ballon, reconnaissance des adversaires).

Nous n'avons pas développé de travaux directement sur ce problème, nous nous sommes concentré sur le problème des robots fourrageurs dont les trois caractéristiques essentielles le différenciant sont :

- une distribution géographique de *plusieurs éléments* à traiter,
- la non connaissance à la fois de l'environnement et de la position des éléments à traiter,
- un système composé d'agents uniquement *coopératifs*.

### 8.1.3 Travaux liés aux robots fourrageurs

L'étude du problème des robots fourrageurs et de ses variantes a donné lieu à de nombreux travaux théoriques et expérimentaux ces dix dernières années. Les thèmes traités sont plus précisément : la résolution *distribuée* de tâches telles que le foraging, consuming et grazing (cf. définitions dans le chapitre 6), les *communications* inter-agents (implicites et explicites) et l'étude de l'impact du nombre d'agents sur les solutions distribuées.

Les résultats les plus marquants peuvent être résumés par les travaux suivants :

- R. Brooks est le premier à avoir proposé l'envoi de nombreux robots simples et autonomes sur des planètes voisines [Brooks et al., 1990]. Il est à la base du principe de l'architecture de subsomption [Brooks, 1986].

- L. Steels et J.L. Deneubourg sont à l'origine des approches émergentistes et d'inspirations éthologiques [Deneubourg et al., 1986] [Steels, 1989]. Ces travaux ont donné lieu à la définition du problème des robots fourrageurs, ainsi qu'à l'idée du dépôt de marques pour constituer des chemins dans l'environnement [Steels, 1989].

En collaboration, ils ont réellement développé des robots collectifs capables de traiter des problèmes comme celui du tri d'éléments [Deneubourg et al., 1991]. Enfin, notons que J.L. Deneubourg et S. Goss ont proposé une solution au problème de fourrage consistant à utiliser les robots comme "bornes" pour former des chemins [Goss and Deneubourg, 1991]. Cette technique est présentée plus loin.

- A. Drogoul et J. Ferber ont analysé en détail l'approche par dépôt de marques

consistant à créer des chemins dynamiques entre les sources de minerai et la base. Cette solution, d'inspiration éthologique (fourmis), est présentée en détail ci-après [Drogoul and Ferber, 1992]. A. Drogoul a aussi proposé une architecture réactive permettant la spécialisation des agents par adaptation de leur comportements aux besoins de l'environnement (Etho-Modelling Framework [Drogoul, 1993]).

- M.J. Mataric a réalisé les premières expériences de résolution collective avec des robots mobiles autonomes [Mataric, 1994a], traitant avec succès des tâches de foraging, grazing, consuming et maintien en formation. Avec M. Fontan elle a mis en évidence les problèmes engendrés par un trop grand nombre d'agents sur la division territoriale des tâches [Fontan and Mataric, 1996]. Depuis, avec son équipe, elle poursuit des travaux de développement de robots mobiles possédant des capacités d'apprentissage, pour le traitement de tâches de types box pushing et foraging [Mataric, 1997] [Goldberg and Mataric, 1999] [Mataric, 2001].

- R.C. Arkin a proposé très tôt une architecture de robot mobile autonome réactif (schema-based control architecture) [Arkin, 1987] permettant de traiter des tâches de navigation et de foraging sans communications. Depuis il poursuit des travaux, en collaboration avec T. Balch, sur des problèmes de navigation de véhicules autonomes en formation et de conception de robots mobiles individuels ou collectifs [Arkin, 1998] [Arkin and Balch, 1998] [Stoytchev and Arkin, 2001].

## 8.2 Application de l'architecture aux robots fourrageurs

### 8.2.1 Traitement distribué du problème

L'intérêt de l'étude du problème de fourragement ne réside pas dans la recherche d'une solution optimale en temps, mais de solutions garantissant avant tout des qualités de *robustesse* (résistance aux pannes, à la dégradation des communications, aux conditions variables de l'environnement), *d'organisation* et de *faible coût*. Généralement, le problème est abordé par une approche complètement décentralisée faisant intervenir de nombreux agents identiques et sans connaissance de l'environnement. Nous avons repris ce pré-requis auquel nous avons ajouter l'émission de signaux par les agents.

Avant de justifier notre choix et présenter l'intégration du modèle satisfaction-altruisme, nous énonçons le problème tel que nous l'avons étudié et simulé :

---

**Problème des robots fourrageurs**


---

- soit un environnement comportant des sources (de minerai), dont le nombre et les positions sont inconnues, et qui contiennent une quantité *finie* de matériau.
- l’environnement est parsemé de nombreux obstacles fixes de formes quelconques<sup>2</sup>,
- la base des robots est située dans cet environnement, elle émet en continu un signal permettant de toujours la localiser et de limiter l’aire d’exploration des robots,

Question : comment définir et organiser  $N$  agents/robots identiques pour découvrir toutes les sources accessibles, les exploiter totalement et ramener tout le matériau à la base ?

---

### 8.2.1.1 Principe de résolution distribué

Les trois comportements généraux d’un agent fourrageur sont l’exploration, l’exploitation (ou consommation) et le transport du matériau vers sa base. Plus précisément ces comportements peuvent s’exprimer par l’algorithme suivant :

---

1. état initial : les robots sont à la base
  2. explorer *aléatoirement* l’environnement, si *découverte* d’une source passer à l’étape 3, si *niveau d’énergie trop bas* alors rentrer à la base se ravitailler
  3. *exploiter* la source, si la *benne est pleine* alors passer à l’étape 4, si la source est épuisée s’éloigner de la zone puis reprendre l’étape 2
  4. la *benne pleine* déclenche un retour à la base pour y décharger le matériau, puis reprise à l’étape 2
- 

Cet algorithme est le plus simple que l’on puisse trouver pour la résolution distribuée de ce problème (introduit par [Steels, 1989]). Il n’utilise aucune communication, aucune carte, aucun apprentissage et aucune marque sur l’environnement. Cette simplicité lui confère des qualités de robustesse individuelle et globale : robustesse de chaque agent (et faible coût) et résistance de la solution à d’éventuelles pannes d’agents.

De plus, le déplacement aléatoire sur un espace *fini* garantit l’exploration en un temps fini de tout l’environnement ([Doncker et al., 1998]), donc la découverte et le traitement de toutes les sources.

Cependant, il s’agit certainement de l’algorithme distribué le moins efficace en temps. Les deux raisons essentielles à cela sont (i) *l’absence d’échanges d’informations entre les agents* et (ii) l’exploration réalisée par une simple marche *aléatoire*.

---

<sup>2</sup>Dans les simulations, par commodité, nous avons défini les obstacles par composition de formes rectangulaires.

Les travaux menés depuis dix ans sur cet algorithme ont consisté à introduire des raisonnements et/ou des communications pour tenter de réduire les durées de résolutions. Ces diverses approches sont présentées ci-dessous.

### 8.2.1.2 L'exploration

Effectuer une exploration de l'environnement par une méthode plus pertinente que la marche aléatoire est possible. Il s'agit de processus de planification exigeant une connaissance de l'environnement (ce qui est exclu ici) et des communications de hauts niveaux. Cependant, l'ensemble de ces contraintes impose souvent une vitesse d'exécution lente, rendant l'approche inadaptée à des environnements dynamiques.

Aussi, nous nous intéressons essentiellement à des approches collectives mêlant aléatoire et repérages dans l'environnement. Ces différentes approches sont les suivantes :

**Découper l'espace** : L'environnement étant initialement inconnu, le concepteur des agents peut adopter l'une des deux stratégies suivantes : (i) autoriser tous les agents à parcourir l'intégralité de l'environnement ou (ii) décomposer l'environnement en zones fixes et limiter leur exploration à certains agents. Cette deuxième solution n'est généralement pas retenue car elle pose les problèmes du *découpage* d'un environnement *inconnu*, du *repérage* de ces zones par les agents et enfin parce qu'elle casse le principe de coopération entre tous les agents (cf. exemple en simulation [Fontan and Mataric, 1996])

**Apprentissage de l'environnement** : En construisant une *carte*, l'agent peut repérer les zones qu'il a déjà exploré et noter la présence de sources ou de tout autre élément pertinent de l'environnement. Cependant, cette solution est très complexe à mettre en oeuvre (elle fait l'objet de nombreuses recherches, cf. [Oriolo et al., 1998]) et elle résiste mal aux environnements dynamiques et/ou sans repères (il suffit de s'imaginer des environnements forestier, sous-marin ou de travaux publics).

**Marquer l'environnement** : C'est une solution doublement intéressante car elle permet à la fois de *repérer* les zones pertinentes/attractives et de *communiquer* indirectement des informations entre agents. Les algorithmes les plus intéressants de cette approche sont les suivants :

1. Inspirés par l'exemple des fourmis fourrageuses, l'idée de marquer l'environnement pour des robots a été introduite par L. Steels [Steels, 1989] et a été ensuite étendue par

A. Drogoul [Drogoul, 1993]. Le principe général est assez simple, les agents déposent des “miettes” sur leur parcours lorsqu’ils reviennent d’une source de minerai. Ainsi, ils constituent des chemins entre les sources et la base. Tous les agents en exploration peuvent utiliser ces pistes pour retrouver les sources découvertes. Pour permettre à ces chemins de disparaître une fois qu’ils ne sont plus utiles (sources épuisé), les agents appliquent une rétroaction négative lorsqu’ils les remontent, ce qui consiste à ramasser une miette par unité de déplacement. La meilleure “version” de ces robots marqueurs est obtenue pour *deux* miettes déposées par unité de déplacement pendant le retour à la base et *une* seule miette ramassée lorsqu’ils suivent les chemins. La rétroaction positive est privilégiée, rendant possible le recrutement, tout en assurant la disparition des marques.

Pour traiter les problèmes d’embouteillages survenant autour de la base et des sources, ce modèle est complété par une action d’échange de matériau entre agents. Chaque agent est capable de *détecter* ses voisins chargés de matériau et de réaliser un *transfert* de celui-ci entre sa benne et celle de son voisin. Ainsi, il *émerge* un processus coopératif très efficace : la *chaîne* de transport. Les encombrements ne sont plus pénalisants, car les agents immobilisés et côte à côte constituent des chaînes de transport qui transfèrent, par échanges locaux, le matériau des sources vers la base [Drogoul and Ferber, 1992].

Aussi performante et adaptative que soit cette solution en simulation, il faut s’interroger sur sa faisabilité en robotique. Deux problèmes majeurs la rendent relativement difficile à implémenter : (i) ces processus de marquages nécessitent un grand nombre d’agents (notion de masse critique) et chaque agent doit posséder un nombre important de marques (de plus, la nature même de ces marques pose problème : elles doivent être détectables, récupérables et résister aux déplacements de nombreux robots), (ii) les tâches mises en jeu dans cette solution sont pour des robots réels très coûteuses en temps et en complexité de manipulations [Doncker et al., 1998] : le dépôt des marques et leur récupération est un problème technique non-trivial et potentiellement consommateur de temps, le transfert de tout le matériau d’une benne à l’autre peut lui aussi être très long et coûteux en énergie.

2. Pour pallier le problème du dépôt et du ramassage des marques, S. Goss et J.-L. Deneubourg ont proposé une méthode permettant aux robots de servir eux-même de bornes pour constituer des chemins [Goss and Deneubourg, 1991]. L’intérêt d’une telle démarche est de produire des chemins par segments (deux robots fixes communicants) qui contournent naturellement les obstacles. Toutefois, cette solution est très coûteuse en agents, car ceux-ci sont “paralysés” longuement dans leur rôle de balise.

3. S. Doncker, S. Regnier et D. Duhaut ont proposé une variante des deux méthodes précédentes permettant d'éviter à la fois le marquage de l'environnement et la paralysie des agents [Doncker et al., 1998]. Comme dans la technique précédente, les agents servent eux-même de "marques" pour créer des chemins entre les sources et la base, mais ils peuvent aussi participer au transport du matériau. Les agents sont toujours dans l'un des états suivants : soit en exploration, soit sur un "chemin" (un agent perçoit toujours la direction de la base et se souvient de la direction d'une source découverte). Le comportement d'un agent découvrant une source consiste à osciller entre cette source et la base. Tout autre agent le rencontrant est "recruté" pour le parcours du chemin : ce lieu de rencontre devient la nouvelle origine d'oscillation pour l'un est la destination pour l'autre. Ainsi, les agents oscillent chacun sur une portion réduite du chemin. S'ils sont suffisamment nombreux, les agents vont converger vers la constitution de chaînes fixes entre sources et base. Comme pour les précédentes méthodes, cette solutions est simple donc robuste et adaptative aux variations de l'environnement, mais elle nécessite aussi de nombreux agents. De plus, toute erreur d'angle dans les déplacements provoque la non rencontre des agents et l'éventuelle perte de chemins.

### 8.2.1.3 Les communications

L'utilisation de communications de hauts niveaux se prête mal au traitement du problème des robots fourrageurs. La première raison est d'ordre technologique. La nature même de ce problème est d'être géographiquement distribué, très dynamique et peuplé de nombreux agents mobiles. Ainsi, les communications entre robots sont nécessairement délicates à établir sans un taux de pertes élevé.

La seconde raison freinant l'approche planification/communications est la complexité naturelle du problème due à la non connaissance de l'environnement et à sa taille potentiellement très vaste. Ainsi, les algorithmes vraiment efficaces pour ce type de problèmes relèvent plus de l'organisation géographique et de la répartition des tâches que de la résolution par raisonnement symbolique.

Notre démarche est d'introduire, par le modèle satisfaction-altruisme, des communications simples (signaux attractifs et répulsifs) permettant d'assurer à la fois la navigation individuelle des agents, la résolution du problème et l'emploi de stratégies coopératives, mais sans utiliser de processus de planification ni de connaissances préalables sur le système.

### 8.2.2 Vers des robots fourrageurs altruistes

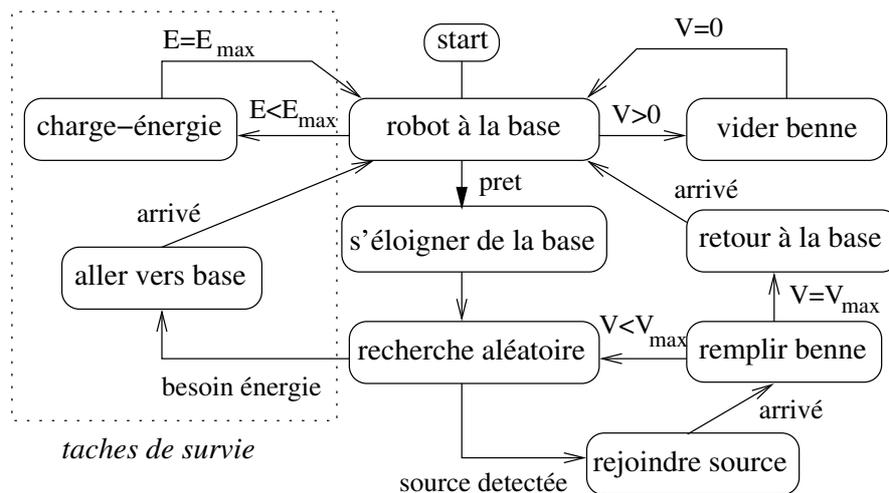
Notre traitement du problème du fourragement s'appuie donc sur l'algorithme *de référence* mais complété des capacités proposées par le modèle satisfaction-altruisme. L'objectif étant d'obtenir une solution à la fois *plus efficace en temps* que l'algorithme de référence, mais qui conserve ses qualités initiales de *robustesse* et d'*adaptation* aux variations de l'environnement. Nous montrons que l'emploi de simples signaux locaux<sup>3</sup> permet de garantir ces qualités de réactivité, de robustesse et d'adaptabilité.

Notre étude se base sur un ensemble de simulations et de mesure des performances de résolution du problème. Le système simulé est un ensemble homogène d'agents situés exécutant un comportement purement réactif complété d'actions de coopérations (signaux et comportements altruistes).

Chaque robot est défini initialement par les capacités suivantes :

- mobilité omnidirectionnelle, déplacement à vitesse constante,
- perception des obstacles dans la direction de déplacement, sur un secteur d'ouverture  $[-\Theta^\circ, +\Theta^\circ]$  et de rayon  $R_p$ ,
- autonomie en énergie limitée (nécessité de rentrer à la base pour recharger les batteries),
- possibilité d'extraire du matériau des sources et de retourner à la base.

Nous en avons déduit le comportement générique (classique) de chaque agent, exprimé par le graphe d'état suivant :



Dans la section suivante, nous implémentons et testons ce comportement de référence

<sup>3</sup>L'aspect local des signaux est étendu vers une notion de portée plus vaste.

à travers le modèle proposé. Puis nous intégrons successivement : l'émission de signaux attractifs ou répulsifs lors de la découverte des sources, l'attente aux sources (ou passage de relais), la propagation des signaux et enfin la possibilité de faire varier la portée de ces signaux.

## 8.3 Simulations, résultats et analyses

### 8.3.1 Les simulateurs

Nous avons développé des outils permettant de simuler l'évolution de robots mobiles en environnements contraints. Nos objectifs étaient multiples :

- vérifier les capacités de résolution et d'auto-organisation du modèle,
- régler/étudier les divers paramètres,
- étudier les phénomènes de propagations des signaux,
- étudier la navigation des agents, leurs trajectoires.

Ces différents objectifs nous ont conduit à développer deux variantes du même noyau de simulation. La première étant orienté vers le traitement du problème de fourragement par de nombreux agents sur de vastes environnements. La seconde, au contraire, se concentrant sur la navigation individuelle, les conflits spatiaux et leur résolution (chapitre 9) et sur l'étude de systèmes hétérogènes (chapitre 10).

Les outils développés s'exécutent sur la plate-forme de simulation Madkit [Gutknecht and Ferber, 2001], qui propose un ensemble de classes abstraites d'agents, de moyens de les faire communiquer et de structures d'organisations (agent-groupe-rôle : le modèle Alaadin [Gutknecht, 2001]). Cependant, le simulateur que nous avons développé utilise seulement une librairie de classes spécialisées dans l'exécution d'agents situés sur un environnement abstrait (la ReactiveLib).

Dans notre étude, l'apport majeur de Madkit est de nous avoir donné la possibilité de gérer(/tester) librement l'ordre d'exécution des actions des agents et de l'environnement (le scheduling). Nous avons ainsi mis en place le *cycle* de simulation donné figure 8.2.a.

A partir de cette séquence, nous avons défini et programmé les comportements des agents, les échanges de signaux, des modèles de l'environnement et enfin leurs représentations graphiques respectives. Madkit nous a permis aussi de développer facilement plusieurs représentations graphiques de l'exécution d'une même simulation et d'intégrer des sondes affichant des informations sur le système.

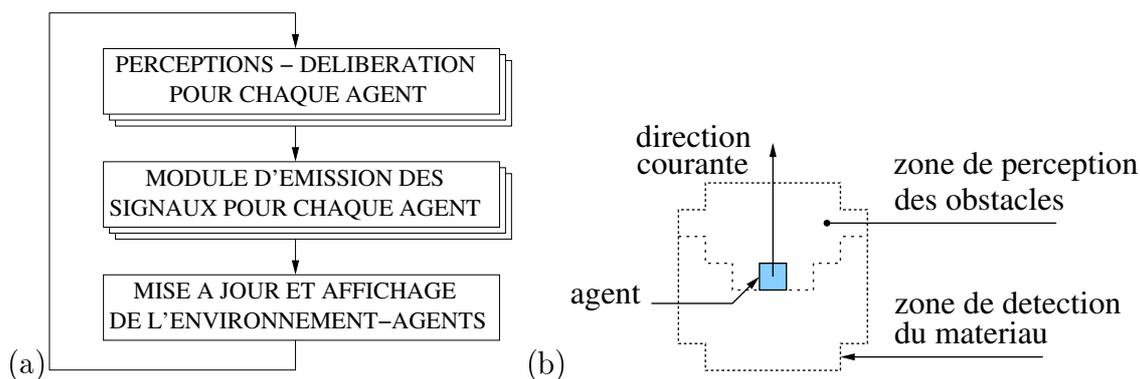


FIG. 8.2 – (a) Cycle de simulation (b) perceptions d'un robot simulé

### Comparaison de plusieurs modèles

Nous avons simulé et comparé cinq variantes de comportements des agents : (1) l'algorithme de référence (qualifié de réactif pur) (2) le modèle avec signaux et altruisme mais sans propagation des signaux (3) ce dernier modèle avec l'introduction de la notion d'attente (4) l'ajout de la capacité de propagation des attractions et enfin (5) l'adaptation dynamique de la *portée des signaux* (sans propagation).

### 8.3.2 Implémentation de l'algorithme de référence

#### Capacités extérior-perceptives des robots

Chaque agent dispose d'un détecteur du matériau recherché, dont la zone de perception est un disque de rayon paramétrable. L'agent perçoit devant lui la présence d'obstacles, dans un secteur d'ouverture  $[-45^\circ, +45^\circ]$ . La figure 8.2.b représente ces différentes perceptions pour un rayon de perception de 3 pixels autour de l'agent.

#### Codage des fonctions (ou comportements)

Le graphe d'état, du comportement générique d'un robot fourrageur (donné section 8.2.2), s'exprime dans notre modèle par l'ensemble des conditions<sup>4</sup> et fonctions suivantes (Motiv. est le poids de la fonction à son déclenchement, après il vaut  $P(t)$ ) :

<sup>4</sup>les conditions sont représentées ici sous forme (de conjonction) de prédicats unaires.

Perceptions / Conditions d'actions	Comportement	Motiv.	n
lieu(base) & benne(vide)	s'éloigner de la base	0.5	1
batteries(faible)	rentrer à la base	surv	2
lieu(base) & batteries(faible)	charger batteries	surv	3
$\neg$ lieu(source) & détection(source)	aller vers source	0.6	4
lieu(source) & $\neg$ source(vidé) & $\neg$ benne(pleine)	remplir benne	0.8	5
benne(pleine)	rentrer à la base	1	6
lieu(base) & $\neg$ benne(vidé)	vider benne	0.8	7
Si aucune situation précédente	marche aléatoire	0.1	8

Rappelons qu'en cas d'échec d'une fonction, celle-ci est temporairement inhibée. Dans le cas de la fonction 5 de remplissage à une source, l'échec (pouvant être du à l'épuisement de la source) inhibe aussi la fonction 4, ceci afin d'éviter une oscillation entre départ et attraction de la source vide.

**Calcul des satisfactions personnelles :** La satisfaction personnelle de chacun des comportements est calculée à chaque itération par l'une des trois mesures suivantes ( $P_{max}=1$ )

- pour toute tâche de déplacement :

$$\begin{cases} \text{si } dep. \text{ impossible} & P(t) = \max(P(t-1) - 3.dS, -1) \\ \text{sinon} & P(t) = \min(P(t-1) + dS + 2.dS.\cos(\theta), 1) \end{cases}$$

ou  $\theta$  est l'angle entre la direction du but courant de l'agent et la direction effectivement réalisée,  $dS$  l'unité de variation de la satisfaction (=0.01).

- chargement du matériau :

$$\begin{cases} \text{si } source \text{ non } epuisee & P(t) = \min(P(t-1) + k.dS, 1) \\ \text{sinon} & P(t) = \max(P(t-1) - k'.dS, -1) \end{cases} \quad (k' \gg k)$$

- les fonctions de vidage de la benne à la base et de recharge des batteries n'ont pas été simulées avec des contraintes éventuelles, ainsi durant leur exécution  $P = 1$ .

### Coefficients de sélection et de combinaison

- $\alpha = 0.7$ , nous débutons l'étude avec une valeur volontairement supérieure à 0.5 pour s'assurer d'un comportement coopératif des agents. Sa variation est évoquée au cours de l'étude, nous y revenons au chapitre suivant.
- $\gamma = 0.3$ , renforce le poids de la tâche en cours pour créer une légère persistance, cette valeur est choisie suffisamment petite pour ne pas empêcher toute fonction d'être

déclenchée lorsque l'agent est en exploration :  $\forall f_i M_{i \neq \text{explo}} - \gamma > M_{\text{explo}} (= 0.1)$ .

- Les agents se déplacent à vitesse constante, ainsi nous avons transformé la somme des deux vecteurs but et évitement des obstacles (de pondération  $g_1$  et  $g_2$ ) en une variation de la direction courante de l'agent (cf détails annexe 14.1.2). Dans l'environnement considéré, la faible densité d'obstacles nous permet de négliger le troisième terme de l'équation 6.3 ( $g_3=0$ ).

Les autres paramètres relatifs aux robots et à la simulation sont donnés en annexe 14.1.

### 8.3.2.1 Simulations

Le premier outil de simulation permet de représenter dans une matrice (de pixels) des environnements contenant des obstacles de formes quelconques, où chaque agent a la taille d'un pixel et son affichage se fait par la représentation de son vecteur direction (optionnellement d'autres vecteurs de types évitements peuvent être affichés).

Leur représentation graphique est soit en vue de dessus simple, soit en vue 2D-iso "relief". Cette dernière a été développée dans le but d'afficher une représentation "3D" des signaux de satisfactions interactives émis par les agents (cf. plus loin), mais aussi de leur satisfaction personnelle (cf. [Simonin, 2001]).

### Premiers résultats

Le modèle a été testé sur différents environnements (définis aléatoirement ou par éditeur). L'ensemble des simulations effectuées montre une résolution systématique et attendue du problème. Deux observations générales :

- le module de navigation des agents calcule des trajectoires non bloquantes, combinant la direction d'un but avec l'évitement des obstacles fixes ou mobiles (autres agents),
- le module de sélection d'action et d'évaluation de la satisfaction personnelle assure un comportement cohérent à chaque agent (cf. section 8.3.4), qui mène à la résolution du problème et à la gestion de l'énergie de chacun.

Nous ne nous sommes pas intéressés au problème de la terminaison de la résolution (problème inhérent à une exploration aléatoire [Doncker et al., 1998]), elle est détectée par le simulateur lorsque toutes les sources sont épuisées (dans un cas réel, on pourrait envisager de stopper la sortie des robots lorsque la quantité de matériau ramenée est nulle pendant une longue période de temps).

Nous avons pu constater aussi qu'en présence d'obstacles de formes concaves, certains agents peuvent être piégés dans des minima locaux (attraction par la base, par une source). Dans ce cas, le temps de résolution est plus long en fonction du nombre d'agents piégés.

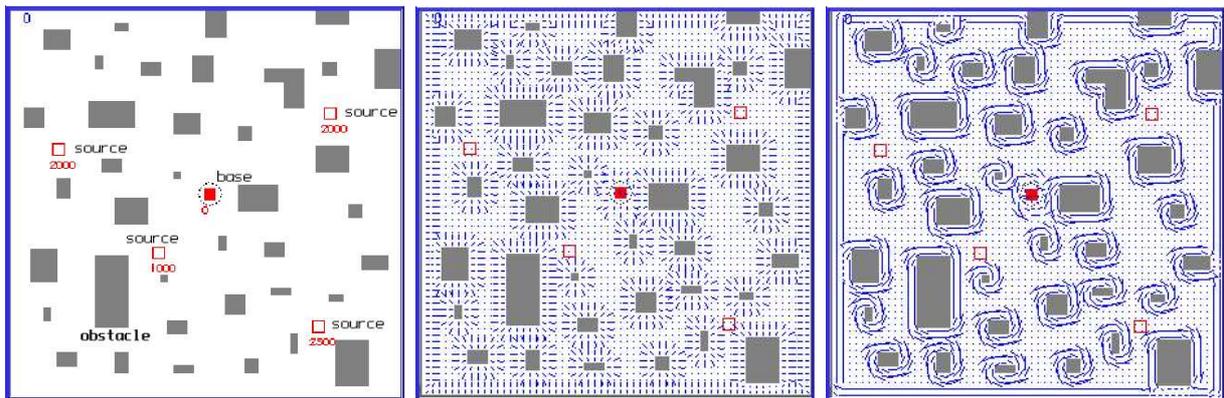


FIG. 8.3 – Environnement d'étude avec représentation des champs vectoriels répulsifs et glissants, calculés en chaque point d'une grille de granularité  $5 \times 5$  pixels, pour une perception locale définie par un disque de rayon  $\simeq 10$  pixels.

Des solutions à ce problème existent, par exemple l'utilisation de comportements de suivi de murs (cf. section 6.4.2). Mais nous avons opté dans cette étude pour des environnements exempts de ce type d'obstacles (l'ajout d'une fonction supplémentaire alourdirait le modèle générique). Cette hypothèse nous garantit la résolution systématique du problème, quelque soit la distribution des obstacles, s'il existe au moins un chemin entre chaque source et la base.

Pour mener une étude comparative de variantes du modèle satisfaction-altruisme avec le modèle réactif pur, nous avons défini un environnement test représentatif du problème.

### Choix d'un environnement d'étude

Paramètres de l'environnement :

- Dimensions :  $300 \times 300$ ,
- 40 obstacles de formes rectangulaires : 12% de la surface,
- une base au centre de l'environnement et quatre sources (position ; vol. matériau) :  
 $(109,183 ; 1000)$ ,  $(32,103 ; 2000)$ ,  $(241,75 ; 2000)$   $(232,240 ; 2500)$ .

Cet environnement est représenté en figure 8.3 en vue de dessus. Sa définition répond à trois critères :

- il est suffisamment dense en obstacles pour tester le principe de navigation proposé,
- il comporte plusieurs sources, proches ou éloignées de la base, dont les volumes de matière sont différents,
- il est suffisamment vaste pour que chaque agent n'en ait qu'une infime perception (soit 0.33%).

Les écrans figure 8.3 représentent l'univers perceptuel des agents. La première copie d'écran

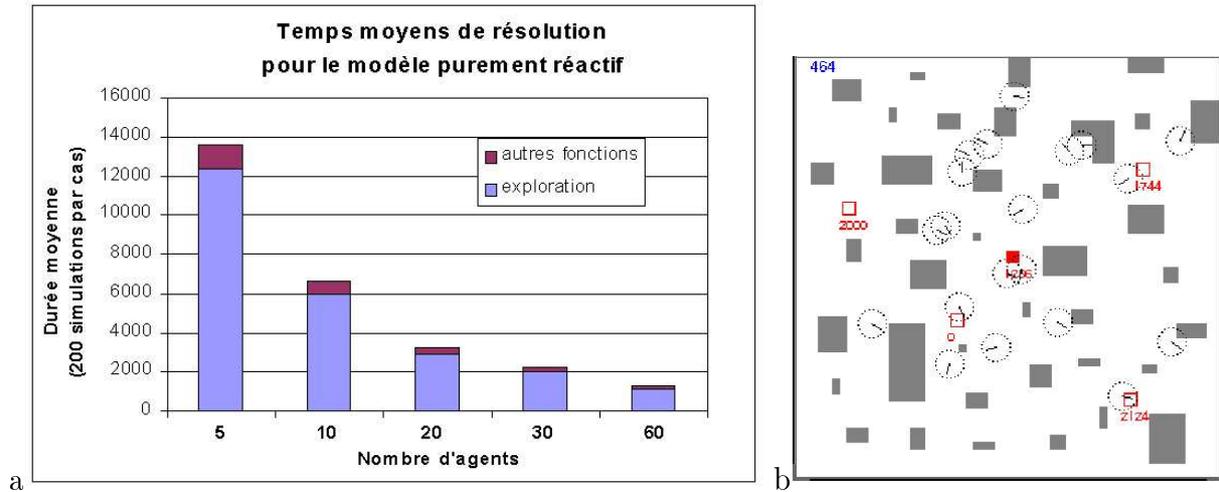


FIG. 8.4 – (a) Temps moyen de résolution en fonction du nombre d'agents (modèle sans signaux), avec pourcentage du temps passé par les agents en exploration par rapport à l'ensemble de leurs fonctions., (b) Copie d'écran d'une simulation avec 30 robots purement réactifs (affichage de la zone de perception des robots).

donne les vecteurs répulsifs déduits de la perception de tout objet dans un disque de rayon 10 pixels (sur une grille de granularité 5x5 pixels). La deuxième copie d'écran représente les vecteurs glissant déduits des répulsions précédentes (orientés ici toujours côté droit). Ces deux représentations ont été obtenues par l'emploi des modules utilisés pour la perception/navigation des robots dans la simulation.

### 8.3.3 Résultats et analyse du modèle sans signaux

Tableau de résultats (chaque valeur est obtenue par moyenne de 400 valeurs, unité = 1 itération) :

r : modèle de référence	5 agents	10 agents	20 agents	30 agents	60 agents
% perception de l'espace libre	1.97 %	3.94 %	7.88 %	11.83 %	23.66 %
Temps moyen de résolution	13554	6622.8	3192.1	2224.9	1270
% de temps en exploration	90.96%	90.89%	90.23	91.10%	91.94%

Les deuxième et troisième lignes de ce tableau sont données sous forme d'un histogramme en figure 8.4.a (la figure 8.4.b illustre l'état d'une simulation en cours).

Première constatation, le temps de résolution décroît lorsque le nombre d'agents augmente. Résultat classique qui s'explique par une augmentation de la probabilité de découverte des sources lorsque le nombre d'agents croît. Toutefois, si nous augmentons le

nombre d'agents vers des valeurs suffisamment grandes pour créer des gênes spatiales, celles-ci provoquent une chute des performances (donc une "remontée" des temps moyens de résolution, cf. [Drogoul, 1993]).

Sur cet espace de 5 à 60 agents, nous pouvons distinguer trois zones :

- Le passage de 5 à 10 agents diminue le temps de résolution de façon très significative (le nombre d'agents double, le temps moyen de résolution diminue de plus du double),
- de 10 à 20 agents la diminution de la durée moyenne de résolution est proportionnelle à l'augmentation du nombre d'agents,
- de 20 à 60 agents la variation de performance s'amortit.

Lorsque le nombre d'agents est faible ( $\leq 10$ ), chaque agent participe "efficacement" à la résolution, ils "se partagent" les zones à explorer avec une faible probabilité de redondance. Au contraire, avec un grand nombre d'agents ( $> 30$ ), la probabilité de redondance d'exploration entre les agents est très forte, donc l'ajout d'un agent au système n'améliore que sous-linéairement les performances.

Reconsidérons ces résultats dans une perspective d'implémentation réelle. Notre objectif est de traiter des problèmes par l'emploi de plusieurs dizaines de robots simples et robustes. Cependant, le coût d'un tel nombre d'agents n'est pas négligeable, même s'il reste individuellement faible. Nous cherchons donc à minimiser deux critères opposés : le coût en robots et le temps de résolution.

Nous nous sommes ainsi concentré sur le problème de la réduction des temps de résolution pour des populations relativement limitées. Plus précisément, l'étude que nous présentons est menée sur des populations de 10, 20 et 30 robots. Ces valeurs sont à la fois "raisonnables" en coût (en regard de l'environnement considéré) et précèdent la zone de fort amortissement des performances.

### Comportements et objectifs

Toutes les mesures de performances ont été accompagné de l'évaluation d'un indice de temps passé par les agents en tâche d'exploration (ou marche aléatoire). Quel que soit le nombre d'agents, la moyenne du temps passé dans cette fonction, par tous les agents, est proche de 91% de la durée totale. Cette valeur reflète l'inefficacité en temps de la stratégie testée, qui est due à l'exploration par marche aléatoire, à l'incapacité des agents à retourner vers une source déjà découverte et à leur impossibilité d'échanger des informations.

La solution que nous proposons ne se base pas sur la construction de chemins ou la mémorisation de directions, ni sur des modèles planifications/communications de haut niveaux, mais sur l'émission de signaux coopératifs dynamiques (le modèle satisfaction altruisme) qui ne font appels qu'aux perceptions locales des agents.

### 8.3.4 Modèle AVEC SIGNAUX (SANS propagation)

Cette première variante du modèle (notée **s** pour émission de signaux) exploite le principe à deux niveaux :

- Émission de répulsions en cas de gênes spatiales entre agents (dans un disque de portée maximum 10 pixels),
- Émission d'une attraction pendant l'exécution normale (satisfaisante) de la fonction de chargement du matériau et émission d'une répulsion en cas de découverte d'une source épuisée. Ces deux types de signaux sont émis sur une portée de rayon  $R_c = 40$ , soit quatre fois le rayon de perception. Cette valeur est choisie volontairement petite car nous ne sommes pas censé connaître le nombre de sources présentes dans l'environnement, mais seulement l'aire de la surface explorable.

Ces fonctionnalités sont intégrées par l'ajout de la réaction altruiste (comportement 9) et par une modification du comportement 5 :

Perceptions / Conditions d'actions	Comportement	Motiv.	signaux	n
[lieu(source) & $\neg$ source(vide) & $\neg$ benne(pleine)] lieu(source) & source(vide)	remplir benne	0.8	<b>Attraction</b>  <b>Répulsion</b>	5
<b>test d'altruisme :</b> $\alpha \cdot  \text{signal ext.}  > (1-\alpha) \cdot \mathbf{P}$	<b>Altruisme</b>	$\alpha \cdot  I_{ext} $		<b>9</b>

Calcul des signaux attractifs et répulsifs :

Les signaux ne sont pas stoppés par les obstacles, ils s'étendent sur tout le disque de rayon  $R_c$  autour de l'agent. Les valeurs émises varient en fonction du temps, leur intensité croît très vite vers le maximum :

Signal attractif (croissant vers 1) :  $I(t) = I(t-1) + k \cdot (1 - I(t-1))$ , avec  $k=0.02$

Signal répulsif (décroissant vers -1) :  $I(t) = I(t-1) - k \cdot (1 + I(t-1))$

En réception, la valeur du signal  $I(t)$  est pondérée par l'intensité de perception (fonction de la distance à l'émetteur, notée  $d$ ). Ce qui s'exprime par la formule  $I_{ext} = I(t) \cdot \left(1 - \left(\frac{d}{R_c}\right)^n\right)$ ,  $n \geq 1$ ,  $1 \leq d \leq R_c$ .

### Représentation "3D" du modèle

Nous avons vu que le principe de comportement des agents se basait sur la sélection et la combinaison de différentes influences perçues en temps réel. Ces deux types d'influences

sont les perceptions d'obstacles éventuellement proches de l'agent et la réception de signaux de satisfaction interactive.

Le calcul de la trajectoire/comportement des agents (eq. 6.3) représente une combinaison de répulsion des obstacles, d'attractions des signaux positifs et de répulsions des signaux négatifs.

Considérons uniquement les signaux de satisfaction, alors les agents évoluent dans un univers localement et dynamiquement composé d'attractions et de répulsions. La force de ces influences est fonction des valeurs transmises ainsi que de la distance entre l'agent et l'émetteur. En considérant la diffusion physique de ces signaux dans le plan  $(X,Y)$  de l'environnement, leur sémantique (les valeurs et leurs signes) peut être représentée par une déformation du plan suivant l'axe  $z$  perpendiculaire (type champ de potentiel).

Pour donner une lecture "physique" à cette projection nous calculons  $z(x,y) = -I_{ext}(t)_{(x,y)}$ , c'est-à-dire qu'en chaque point du plan un signal apparaît par une déformation de la surface d'une valeur opposée à celle qu'il transmet. Ceci dans le but de visualiser les attractions (valeurs positives) comme des "creux" ou des puits et les répulsions (valeurs négatives) comme des reliefs. Ainsi, un agent altruiste peut être vu comme une boule d'acier évoluant sur cette surface déformée dynamiquement par l'évolution des signaux, la réaction altruiste s'apparente alors à la force de gravitation :

- tout agent entrant dans un puits (signal attractif) est attiré en son centre, d'autant plus qu'il s'en rapproche,
- tout agent à la surface d'un relief (signal répulsif) est repoussé vers le bas, d'autant plus s'il est proche du sommet.

Notons que lorsqu'un agent applique une force de contournement, il ne peut plus être vu comme une boule pesante. Dans ce cas, il s'agit plutôt d'un agent "marcheur", bloqué à une certaine altitude, qui est obligé de contourner l'obstacle pour le dépasser.

Ce choix de représentation 3D de l'univers perceptuel et comportemental de l'agent permet d'y intégrer naturellement les obstacles physiques. En effet, ceux-ci provoquent chez l'agent une répulsion ou un contournement (plus exactement une combinaison des deux) comme le font les reliefs représentant les signaux répulsifs. Les obstacles sont donc représentés dans cet espace par un ensemble de reliefs de hauteur constante.

Lorsque cet espace est déformé par une attraction ou une répulsion, les obstacles, de hauteur constante, subissent eux aussi cette variation. Mais ils restent en relief par rapport à la surface. Par exemple pour une attraction : l'agent est attiré par le puits tout en contournant d'éventuels obstacles présents dans la pente.

Pour reprendre l'image de la boule pesante :

- les agents sont attirés par la formation d'un creux dans la surface de l'environnement,
- les agents sont repoussés par les zones s'élevant brusquement,
- les agents contournent les reliefs en combinaison avec les précédents comportements.

Calcul de la surface :

En chaque point  $x,y$  d'une grille couvrant l'environnement, à l'instant  $t$ , sa hauteur  $z(x,y)$  se calcule ainsi :

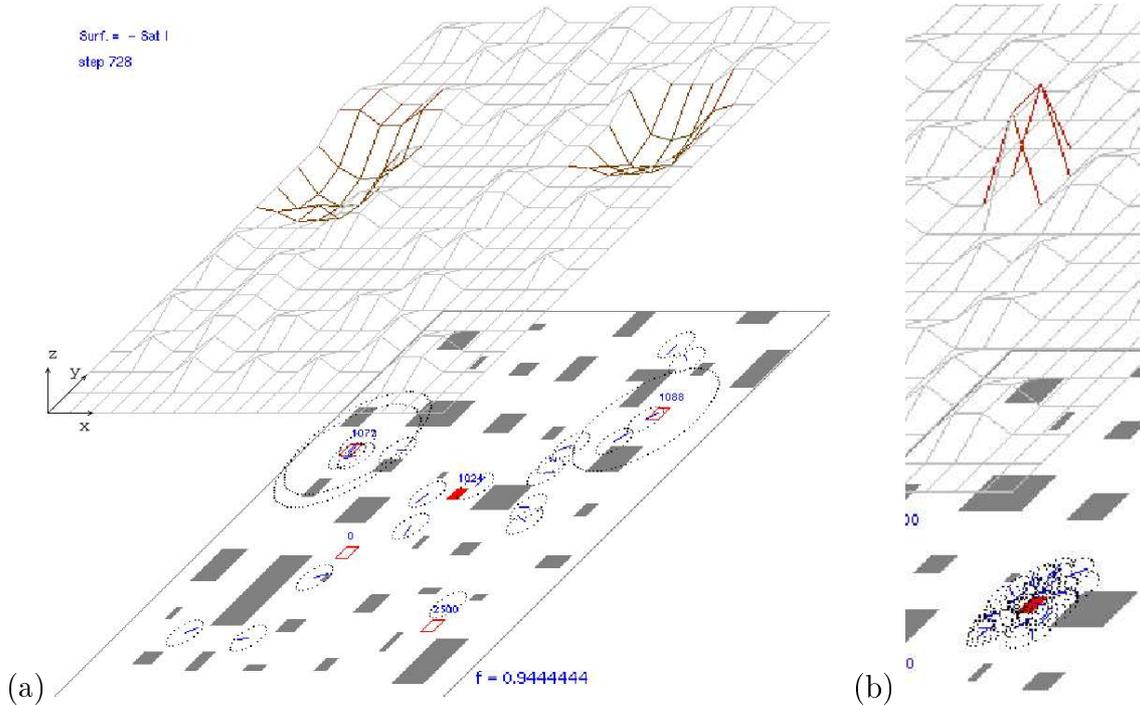
$$Obs(x, y) = \begin{cases} 1 & \text{si } obstacle \text{ en } x, y \\ 0 & \text{sinon} \end{cases}$$

$$d_{sigI_k(x,y)} = \sqrt{(x_{sourceI_k} - x)^2 + (y_{sourceI_k} - y)^2} \quad I_j(x, y) = I_j(t) \cdot (1 - d_{sigI_j(x,y)}^2 / R_c^2)$$

$$SigI(x, y) = I_k(x, y) \text{ tel que } \forall j I_k(x, y) \geq \max(I_j(x, y))$$

$$z(x, y) = H \cdot Obs(x, y) - SigI(x, y)$$

$H$  est la hauteur constante des obstacles. Il ne reste plus qu'à faire une projection classique du point  $(x,y,z)$  dans la représentation 2D-iso. Les figures ci-dessous donnent des exemples de cette représentation.



Résultats qualitatifs :

En début de simulation les agents sont tous très proches les uns des autres, ce qui provoque un certain nombre de gênes spatiales. En conséquence, des signaux répulsifs sont émis (cf. fig. ci-dessus (b) - 20 agents). Même s'il est très difficile de mesurer l'apport de ce processus de répulsion, son intérêt est très visible lors des simulations : les agents se répartissent dans l'environnement de façon très homogène (cf. [Simonin and Ferber, 2000]).

Le processus d'exploration mène à la découverte de sources, les agents se chargent en matériau tout en émettant des signaux attractifs (cf. fig. ci-dessus (a) ).

Le modèle de navigation fonctionne de manière efficace : les agents combinent évitement des obstacles, des robots, attractions vers les sources, et ils font une sélection entre signaux répulsifs et signaux d'appels.

Toutefois, un problème prévisible apparaît, des agents se trouvent piégés temporairement dans les puits d'attractions, alors qu'on souhaiterait qu'ils transportent leur benne pleine jusqu'à la base. Cette situation se produit lorsque plusieurs agents se situent autour d'une source et qu'au moins l'un d'eux émet un signal d'appel, dans ce cas il empêche le départ des autres. Mais, cette situation n'est que de courte durée, puisque dès que chaque robot présent à la source a rempli sa benne (ou que la source est épuisée) alors aucun signal d'appel n'est émis et les agents peuvent rentrer à la base.

Résultats quantitatifs :

s : algo. avec signaux / altruisme	10 agents	20 agents	30 agents	60 agents
Temps moyen de résolution	6610.5	3166.9	2185.8	1061.2
% de temps passé en exploration	89.36%	85.70 %	85.78 %	78.89%
amélioration sur le modèle <b>r</b>	0.2 %	0.8 %	1.76 %	16.5 %

L'amélioration des performances est faible, mais elle croît logiquement avec le nombre d'agents. La probabilité de recrutement par un signal d'appel est directement liée à la densité d'agents dans l'environnement.

Quelles sont les raisons de ces gains si faibles ? Intuitivement, on pourrait supposer que la portée des signaux n'est pas assez importante, pourtant l'augmenter ne changerait rien aux performances :

Un agent n'émet un signal d'appel que pendant la durée du chargement, soit pendant  $\frac{V_{benne}}{Debit_{chargement/it.}} = \frac{250}{8} = 32$  itérations. C'est-à-dire qu'un agent détectant un signal d'appel a au plus 32 itérations pour atteindre la source (la détectée), sinon il repart en recherche aléatoire.

Les phases de recrutements sont donc trop courtes pour être efficaces, sauf si la densité d'agents est très élevée ( $\gg 30$ ).

L'autre solution, que nous développons ci-dessous, consiste à modifier le comportement de l'agent, lorsqu'il se situe à une source, de telle sorte que son signal d'appel soit plus efficace.

### Ne pas perdre les sources découvertes et sortir des zones attractives

L'inefficacité du modèle précédent vient du fait que les agents n'émettent pas assez longtemps les signaux attractifs. Le risque qu'ils repartent sans qu'un autre agent ne soit arrivé à la source est important, et dans ce cas la source est "perdue".

Pour répondre à ce problème, nous considérons que la découverte d'une source est suffisamment importante pour obliger l'agent "découvreur" à stationner à la source jusqu'à l'arrivée d'un autre robot. Durant cette période, il émet en continu un signal d'appel jusqu'à établir un relais avec un autre agent. Cette fonctionnalité est programmée par un état d'attente tant que l'agent ne perçoit pas un autre agent actif à la source. Durant cette attente, la satisfaction de l'agent décroît très lentement, de telle sorte que si l'attente est très longue l'agent abandonnera ce comportement (modélisation d'un état d'impatience). Ce processus est fondamental pour libérer le système d'une éventuelle situation de "famine" (les agents sont tous en attente de l'arrivée d'explorateurs).

Cette solution ne peut être valide que si nous traitons le problème décrit plus haut des signaux attractifs qui piègent les robots autour des sources. Une première solution envisageable est de simplement abaisser la valeur du coefficient d'altruisme  $\alpha$ , de telle sorte que la motivation à rentrer à la base soit plus forte qu'un signal attractif. La tâche de retour à la base a pour valeur de motivation 1.0. Deux agents proches peuvent s'échanger des signaux attractifs de valeur maximale 1.0. Ainsi, il faudrait choisir  $\alpha < 0.5$  pour déclencher le retour et non l'altruisme. Cependant, la satisfaction d'un agent peut fluctuer entre 1 et  $-1$ , remettant en cause la persistance du retour à la base sur l'altruisme.

Plutôt que d'abaisser le coefficient d'altruisme sous la valeur 0.5, donc de risquer de perdre des comportements altruistes, nous considérons que *la fonction d'altruisme peut être inhibée temporairement comme toute autre fonction de l'agent*.

Inhiber la fonction d'altruisme revient à poser temporairement  $\alpha = 0$ . Cette démarche est logique vis-à-vis des *signaux attractifs*, puisque l'agent étant occupé par une tâche particulière il ne peut en réaliser une autre.

Finalement, lorsque l'agent exécute la fonction retour à la base, le déclenchement de l'altruisme pour les signaux attractifs est inhibé. En conséquence, lorsqu'un agent a fini de remplir sa benne et qu'un autre prend le relais, il peut transporter le matériau vers la base

en ne tenant compte que des signaux répulsifs, c'est-à-dire ceux liés à la navigation.

Modification de la fonction 6 :

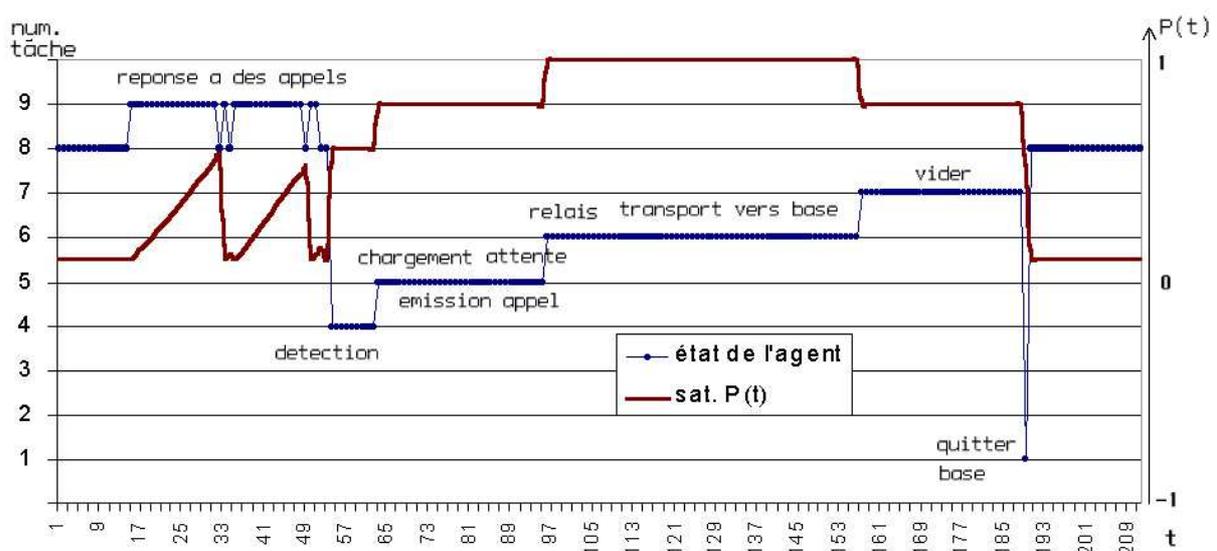
Perceptions / Conditions d'actions	Comportement	Motiv.	signaux coop.	n
[ benne(pleine) & lieu(source) & ¬source(vide) & ¬voisin(charge) ]	Attendre ( $P \searrow$ lentement)	1	<b>Attraction</b>	6
<b>Sinon</b> benne(pleine)	Rentrer à la base & ignorer signaux +	1		6

Résultats des simulations :

a : attente et signaux	10 agents	20 agents	30 agents
Temps moyen de résolution	5960.1	2340	1460.4
% de temps passé en exploration	62.85 %	77.76 %	77.52 %
amélioration sur modèle s	9.8% %	26.1 %	33.2 %

L'amélioration des performances moyennes est très significative, la technique d'attente (ou de relais aux sources) permet d'exploiter complètement le principe des signaux de recrutement.

Le graphique ci-dessous décrit l'état interne d'un agent (sa satisfaction personnelle et les fonctions qu'il a exécuté) enregistré au cours de l'une des simulations. Nous pouvons constater que l'agent a pu exécuter ses tâches sans gênes particulières (la satisfaction est maximale ou en progression pour chaque fonction) et l'agent attiré par un signal attractif (temps 17 à 50) a atteint une source. Il a alors enchaîné les opérations de chargement, d'attente-appel d'un autre agent, puis de transport du matériau à la base.



La méthode n'est véritablement efficace que si le nombre d'agents vaut au moins le double du nombre des sources, afin d'assurer des relais efficaces. On constate d'ailleurs que l'amélioration des performances pour 10 agents est bien inférieure aux cas 20 et 30 agents. Cette analyse est confirmée par les pourcentages de temps passés en exploration, plus faible pour les systèmes à 10 agents, car les robots restent de longues périodes aux sources pour attendre/recruter un agent en exploration.

Le principe du relais aux sources nous permet maintenant d'étudier des extensions du modèle pour réduire davantage les temps de résolution. Nous nous sommes intéressé à deux extensions naturelles du modèle :

- la variation dynamique de la portée des signaux attractifs,
- la diffusion (ou propagation) des attractions entre les agents (principe présenté en section 7.3).

Nous avons simulé ces deux solutions, pour mesurer leur apport respectif et les comparer.

### 8.3.5 Modèle AVEC propagation des attractions

Modification de la fonction d'altruisme :

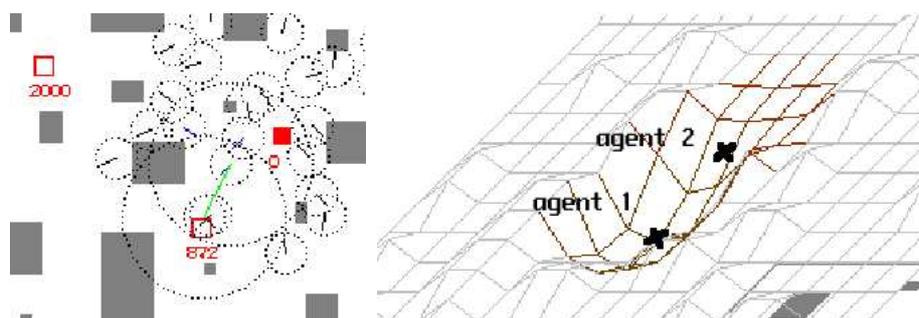
Perceptions / Conditions	Comportement	Mot.	signaux coop.	n
$ \text{signal ext. intensité}  > \text{SatP}$	Altruisme	$\alpha \cdot  I_{ext} $	si capte att. $I_{ext} > I_m$ alors émettre $I_{ext}/2$	9

Paramètre de la simulation :

$I_m = 0.1$ , cette valeur doit être suffisamment faible pour provoquer une propagation quelle que soit la position de l'agent dans le disque d'appel (rappelons que  $I_{ext} = I(t) \cdot (1 - (\frac{d}{R_c})^n)$ ).

Résultats qualitatifs :

Les agents propagent les appels de telle sorte que la zone d'attraction s'agrandit à chaque relais de l'information. La figure ci-après montre une situation de propagation entre un agent situé à une source et un autre près de la base. Ce deuxième émet à son tour un signal attractif. Cette propagation est représentée par deux cercles dans la vue 2D et par une vaste pente entre la base et la source sur la vue 3D.



Résultats quantitatifs :

<b>p</b> : propagation des attractions	10 agents	20 agents	30 agents
Temps moyen de résolution	5133.2	2313.5	1447
% de temps passé en exploration	70.87 %	77.07 %	76.23 %
amélioration sur le modèle <b>a</b>	13.9 %	1.2 %	1 %

Cette technique permet d'améliorer sensiblement les performances précédentes, essentiellement pour la plus petite population. La raison d'un tel résultat est due à la combinaison de deux phénomènes :

- Les phases de recrutement par propagation d'attractions ont lieu rarement, car la surface d'appel n'augmente au mieux que de 50% et se réduit très rapidement de par le rapprochement des agents,
- Le gain en temps d'un recrutement supplémentaire est plus important pour une petite population (où la durée moyenne d'attente d'un agent à une source est très longue) que pour celui d'une population de 20 ou 30 agents (cf. premiers résultats fig. 8.4.a).

Cette analyse est confirmée par l'évolution du temps moyen passé par les agents en exploration. Pour 10 agents, il est passé de 62.85% à 70.87 %, c'est-à-dire que les agents restent moins longtemps en attente aux sources. Pour les populations plus nombreuses, ce pourcentage n'a pratiquement pas varié.

Réflexions autour de cette technique :

- Avantages : La propagation (ou le relais) de signaux attractifs est une solution efficace s'il l'on se place dans la situation où les signaux ne peuvent pas traverser les obstacles (ce qui n'est pas le cas dans les simulations). En effet, les signaux étant relayés par les agents, ils créent des chemins formés de segments ne pouvant couper les obstacles (équivalant à [Goss and Deneubourg, 1991]). Comme les agents évoluent à proximité les uns des autres, ces chemins attractifs empruntent des zones libres, c'est-à-dire contournant les obstacles. Nous allons voir que la dernière technique étudiée ne le permet pas.

- Inconvénients : Le seuil de déclenchement  $I_m$  du relais des attractions est délicat à déterminer. Ce problème peut être contourné en “jouant” sur l’intensité du signal initialement émis, afin de contrôler la distance de propagation. Cette intensité doit être adaptée aux besoins de l’agent pour que l’approche soit pertinente. Par exemple, pour le problème des robots fourrageurs, la détection d’une source riche déclencherait une forte attraction alors que la perception d’une source presque épuisée déclencherait une émission de faible intensité.

Cependant, nous avons précisé que les faibles performances obtenues avec cette technique étaient dues à la couverture trop modeste des appels émis ou relayés par les agents. L’extension de cette limite donne lieu à une dernière variante du modèle, qui est présentée ci-dessous.

### 8.3.6 Variation dynamique des portées d’émissions

L’étude de la précédente technique a montré la nécessité d’adapter la portée des émissions aux besoins des agents. Leur perception de l’environnement étant très limitée, nous utilisons à nouveau leur niveau de satisfaction pour adapter cette portée.

Le processus proposé consiste à doubler le rayon du signal de recrutement lorsqu’aucun nouvel agent n’arrive à la source. Ceci s’exprime par les opérations suivantes :

- La décroissance de la satisfaction est calculée pour que l’attente entre  $P=1$  et  $P=0.5$  dure  $R_c$  itérations. Lorsque la satisfaction atteint la valeur 0.5,  $R_c$  est doublé et le processus reprend (avec  $P=1$ ),
- Cette amplification est limitée par une distance maximale sur la portée d’émission :  $R_{max}$ ,
- Toute arrivée d’un nouvel agent ré-initialise  $R_c$  à sa valeur initiale  $R_{c0}$ .

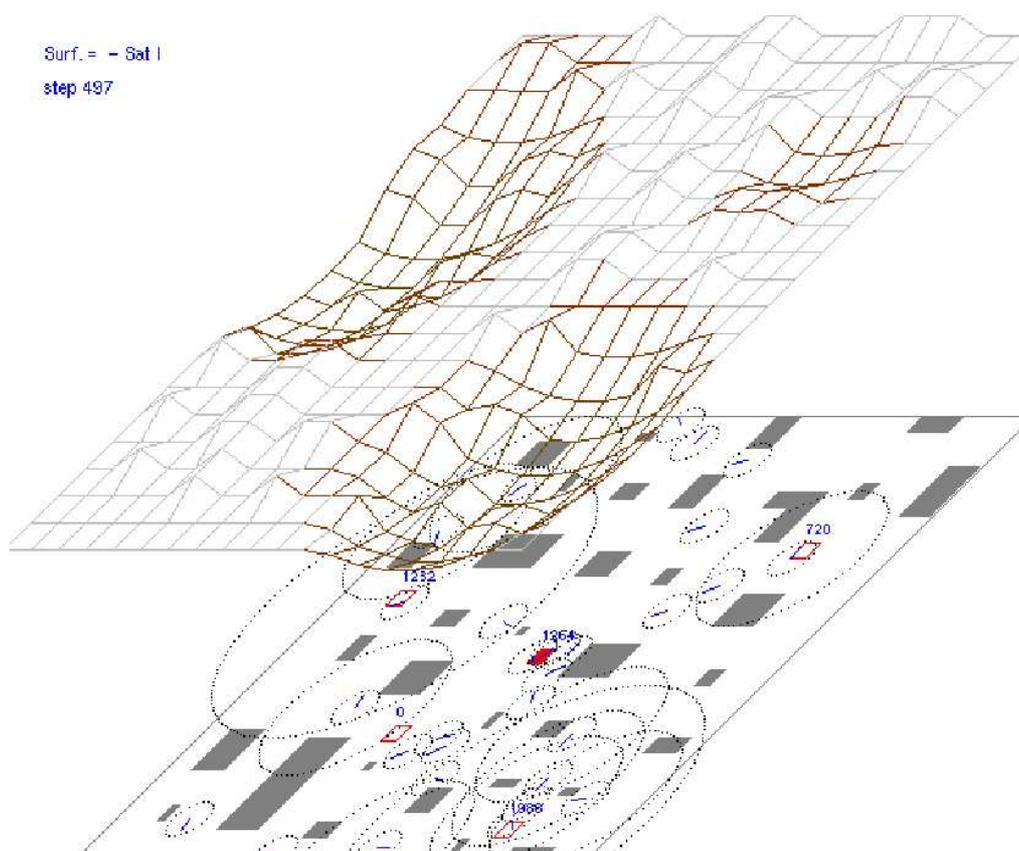
Modification de la fonction 6 :

Perceptions / Conditions d’actions	Comportement	Mot.	signaux coop.	n
[benne(pleine) & lieu(source) & $\neg$ source(vidé) & $\neg$ voisin(charge)] Si $P < 0.5$ & $R_c < R_{max}$	Attendre ( $P \searrow$ de $\frac{1}{2.R_c}$ ) $R_c = R_c * 2$ et $P = 1$	1	<b>Attraction</b> de portée $R_c$	6
<b>Sinon</b> benne(pleine)	rentrer à la base $R_c = R_{c0}$ & ignorer att.	1		6

Paramètres :

$R_{c0} = 40$  et  $R_{max} = 50$  donc la portée d’émission ne peut être doublée qu’une fois.

Résultats des simulations :



$v$ : variation de la portée des signaux	10 agents	20 agents	30 agents
Temps moyen de résolution	2976.3	1570	1024.2
% de temps passé en exploration	62.07 %	68.65 %	66.5 %
amélioration sur modèle <b>a</b>	50 %	32 %	29 %

Les résultats obtenus améliorent fortement ceux du modèle avec attractions à portée constante (de rayon  $R_c = 40$ ). Il est évident qu'en augmentant la portée d'attraction, la surface d'appel croît quadratiquement, rendant tout appel plus efficace.

Cependant cette efficacité provient aussi de l'aspect *adaptatif* de la solution : l'amplification n'est déclenchée que si l'agent est insatisfait de ne voir arriver aucun agent en exploration, et s'il n'en voit pas venir c'est que la densité en agent autour de la source est très faible. Dans ce cas, l'augmentation de la portée de l'appel est justifiée et ne risque pas de provoquer un trop grand nombre de recrutements (comme cela arrivait avec la technique de propagation des attractions). Le comportement de l'agent est bien adapté à sa situation personnelle et à celle de son environnement.

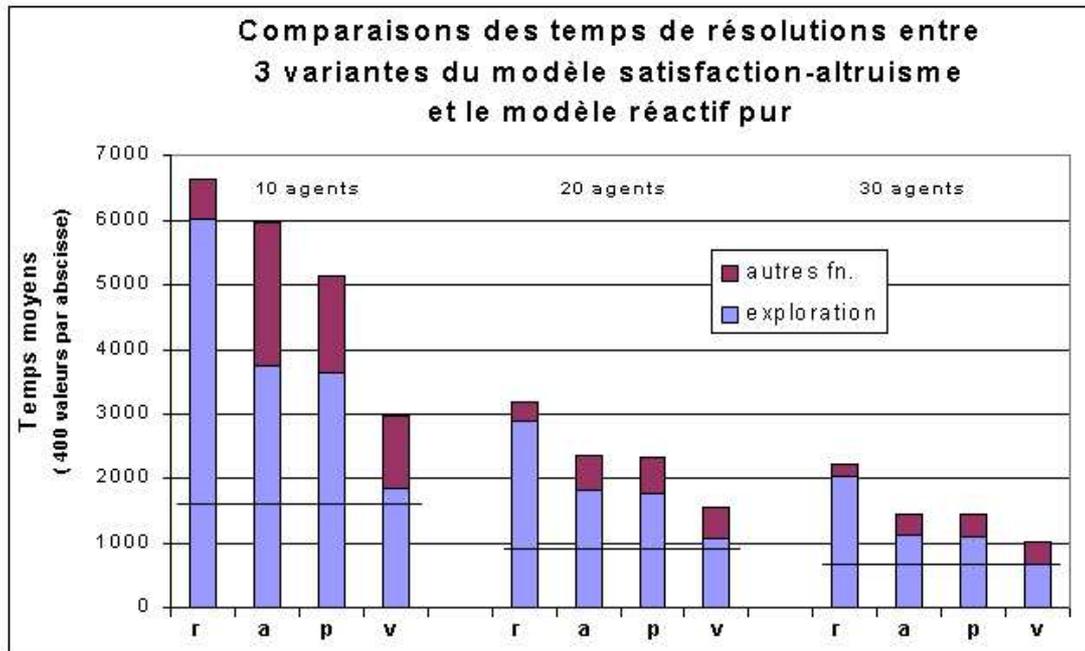


FIG. 8.5 – Bilan de l'étude des différentes variantes du modèle. r : modèle de référence (sans signaux), a : émissions de signaux attractifs et répulsifs, et relais aux sources, p : propagation des attractions entre agents, v : variation dynamique de la portée des émissions (les segments horizontaux donnent la durée minimale théorique).

Réflexions autour de cette technique :

- Avantages : Adaptabilité de la portée des attractions en fonction de l'impatience des agents. L'amplification de ces portées reste constante, contrairement à la méthode de propagation où elle diminue par rapprochement des agents.

- Inconvénients : Cette solution suppose une capacité d'émission à longue distance des agents (dans le cas d'agents situés réels). De plus, s'il existe des obstacles de formes concaves dans l'environnement, ces larges appels peuvent piéger de nombreux agents dans des minima locaux. Au contraire, si les signaux ne peuvent pas traverser les obstacles, l'intérêt de la méthode est réduite, d'autant plus si l'espace est fortement encombré. Il s'agit ici d'une limitation qui n'existe pas avec la solution par propagations.

## 8.4 Bilan

Nous avons évalué le modèle de coopération proposé en étudiant un problème classique exigeant une solution distribuée. Les résultats quantitatifs montrent l'efficacité du modèle, en particulier lorsque les signaux attractifs sont définis en fonction de l'état de satisfaction

des agents. La figure 8.5 résume sous forme d'histogrammes les principaux résultats quantitatifs obtenus sur un environnement d'étude. Ce graphe précise, pour chaque population, la durée théorique minimale<sup>5</sup> que l'on obtiendrait avec un système de planification ayant la connaissance initiale de la position des sources et de leur volume (traits horizontaux). Les résultats obtenus avec la dernière méthode n'excèdent la durée optimale que de 30% (pour une population de 30 agents).

D'un point de vue plus général, le modèle montre son efficacité dans l'auto-organisation des agents. Comme tout système réactif, chaque agent adopte un comportement qui est fonction de ses perceptions locales de l'environnement, générant ainsi une répartition naturelle des tâches. Notre proposition ne remet pas en cause ces caractéristiques, mais y intègre des informations coopératives accélérant certains processus (recrutements lorsqu'une source est découverte, émissions de répulsions si celle-ci est épuisée, propagation des informations).

D'autre part, les situations conflictuelles (ici de gênes spatiales) déclenchent des émissions de signaux répulsifs qui sont immédiatement intégrés aux comportements des agents. Dans les simulations, cela se traduit par une réorganisation locale et rapide de la distribution des agents. Plus généralement, ces simulations montrent la souplesse du modèle comportemental par l'intégration de différentes contraintes : atteindre un lieu, éviter les obstacles et les agents mobiles, persister dans une tâche ou au contraire l'abandonner pour une autre plus importante.

Dans une perspective d'implémentation réelle, il faut noter que l'émission de signaux est relativement simple à mettre en oeuvre, mais surtout que le système réactif à ces signaux est très robuste face à d'éventuelles coupures de transmissions (il n'y a pas de protocole de communication, ni de langage). Ce n'est pas le cas des systèmes cognitifs/délibératifs qui exigent des communications très sûres et qui peuvent être paralysés longuement lorsqu'ils doivent tenir compte de ces bruits ou pannes.

L'environnement test utilisé pour mener l'étude comparative précédente a pour particularité d'être faiblement contraint. Par conséquent, les interférences spatiales n'étaient pas suffisamment nombreuses pour provoquer des gênes durables entre les agents. Le chapitre suivant est consacré au traitement de ce type de situations bloquantes. Pour cela nous avons développé un deuxième outil de simulation permettant de visualiser plus finement les agents et leurs insatisfactions.

---

<sup>5</sup>Cette durée a été calculée en considérant que les agents se déplacent "en ligne droite" entre la base et les sources et qu'ils agissent en parallèle par groupes de taille optimale par rapport au volume de matériau de chaque source.



# Chapitre 9

## Traitement des conflits spatiaux

### 9.1 Introduction

Nous abordons dans ce chapitre les problèmes liés à la navigation de nombreux agents autonomes au sein d'un environnement contraint, c'est-à-dire contenant de nombreux obstacles infranchissables.

Nous nous intéressons aux conflits et aux blocages qui surviennent lorsque la densité d'agents est importante. Ces conflits apparaissent en particulier sur des zones réduites de l'environnement (couloirs, passages étroits, portes, accès à des ressources locales, etc.).

Les agents que nous développons sont par définition *indépendants*, ils n'ont pas de connaissance de la topographie de l'environnement et ne le perçoivent que *localement*. Nous avons vu que ces caractéristiques donnent aux agents des capacités d'auto-organisation et de flexibilité leur permettant de résoudre des problèmes spatialement distribués. Toutefois, ces notions d'indépendance et de localité sont aussi directement responsables de certains conflits spatiaux, car elles empêchent les agents de percevoir ou d'anticiper un ensemble d'actions incompatibles. Ces problèmes étant inhérents à l'approche réactive, nous nous sommes concentré sur la possibilité de les résoudre lorsqu'ils apparaissent.

Nous avons proposé un modèle d'agent situé, à base d'actions réactives et d'émissions de signaux, assurant la navigation individuelle de l'agent, l'émission de signaux coopératifs et l'exécution de réactions altruistes. Ce modèle a été complété d'un processus de propagation des signaux attractifs permettant d'accélérer les recrutements. Mais nous avons vu aussi qu'il existait des propagations implicites (ou indirectes) des signaux répulsifs (chapitre 7). C'est cette dernière notion que nous approfondissons ici pour traiter les problèmes de conflits spatiaux évoqués.

## 9.2 Situations conflictuelles

Lorsque de nombreux agents autonomes évoluent dans le même environnement, ils sont naturellement amenés à se croiser et/ou interagir. Si les agents n'ont pas de mécanisme de coordination ou de négociation, ils peuvent atteindre des situations de conflits pour l'accès à des ressources, et en premier lieu, pour l'accès à l'espace [Fontan and Mataric, 1996].

Les conflits d'accès aux *objets* (fixes) sont généralement traités par des processus de négociation (agents délibératifs, communicants). Cette approche a été adaptée à la résolution des conflits spatiaux, sur des situations relativement spécifiques : passages de carrefours [Wang, 1995b] [Wang, 1995a], de couloirs [Rausch et al., 1995], d'espaces confinés [Qutub et al., 1997]. Toutes ces solutions exigent des agents la connaissance des intentions de chacun et souvent aussi de la topologie de l'environnement. Ce sont des solutions délibératives (centralisées ou hybrides) qui utilisent des communications de hauts niveaux.

Il existe aussi des techniques de navigation réactive mais elles sont généralement adaptées à des problèmes très spécifiques, tels que celui de la simulation du trafic routier (cf. liens [SMARTTEST, 1999], [Hadouaj et al., 2001]) ou de la navigation de robots non communicants [Haninarayan and Lumelsky, 1994].

Récemment, l'équipe de M.J. Mataric a proposé l'utilisation de ce type d'approche locale pour traiter des problèmes de conflits spatiaux entre robots mobiles autonomes [Vaughan et al., 2000]. Ce travail s'appuie sur l'idée de "compétition" locale entre (deux) agents lorsque ceux-ci sont en conflit pour accéder à un lieu de l'environnement. Chaque agent peut essayer de repousser son voisin en lui communiquant un niveau d'agressivité. Celui de plus forte valeur l'emporte et fait reculer son adversaire. Toutefois, à ce jour, cette étude a de fortes limitations :

- le type d'environnement considéré est très peu contraint, il n'y a pas de pièges de type voies sans issues (les agents suivent des lignes au sol),
- le modèle n'a été *simulé* qu'avec un maximum de 4 robots,
- le processus d'interaction n'est pensé que pour *deux* robots,
- le niveau d'agressivité est défini soit par une hiérarchie initiale entre les agents, soit par leur perception de l'espace libre devant eux.

La solution que nous avons développée permet d'aborder les conflits spatiaux sur des populations d'agents de tailles importantes (de quelques unités à plusieurs dizaines d'agents). De plus, nous nous sommes intéressés aux environnements contenant des couloirs étroits, mais aussi des pièges tels que des *corridors fermés (ou impasses)*. Enfin, notre objectif était de proposer une solution simple (réactive) aux problèmes de navigation d'agents indépendants

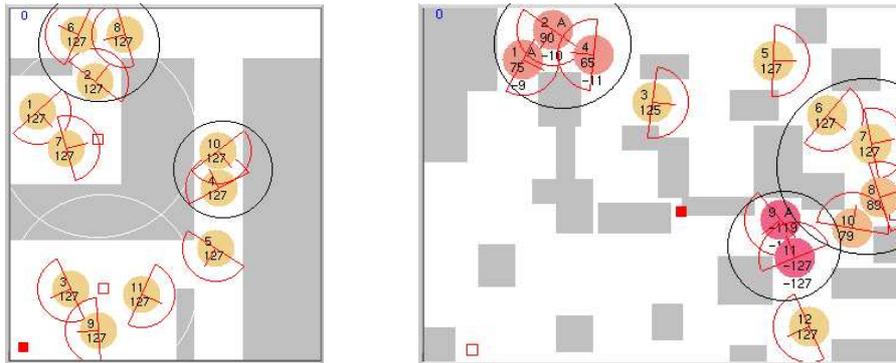


FIG. 9.1 – exemples de conflits spatiaux dans un environnement très contraint (cercles).

évoluant dans des espaces confinés (sans spécialiser les agents à un problème particulier). Pour cela, nous avons utilisé les capacités génériques de l'architecture proposée.

Les conflits d'accès sont d'autant plus complexe à résoudre qu'ils impliquent de nombreux agents. En effet, une situation de blocage peut s'étendre sur une portion importante de l'environnement (plusieurs fois l'aire de perception d'un agent). Nous définissons plus clairement ce type de conflit par l'énoncé suivant :

**Définition** (*conflit spatial*). On appelle conflit spatial une situation où plusieurs agents tentent d'utiliser simultanément un même espace alors que celui-ci est insuffisant pour le permettre.

C'est le cas par exemple pour deux agents qui tentent de se croiser dans un couloir alors que celui-ci est trop étroit pour le permettre. Si les deux agents insistent pour passer, le conflit devient blocage (ou deadlock).

Considérons de façon assez générale un ensemble d'agents mobiles évoluant dans un environnement très contraint (forte densité d'obstacles fixes), cf. fig. 9.1. Inévitablement, des conflits spatiaux apparaissent au niveau des passages étroits. Deux évolutions sont possibles, soit les agents en conflits décident de rechercher un autre passage, soit ils insistent pour passer. S'ils décident de trouver un autre chemin, en supposant qu'il en existe un, le problème peut être résolu. Mais dans le cas contraire, où cette autre solution n'existe pas, les agents restent *a priori piégés*.

Nous allons montrer qu'il est possible de sortir de ces pièges en utilisant l'architecture satisfaction/altruisme définie au chapitre 7.

### 9.3 Principe de résolution

Les signaux permettent aux animaux et aux êtres humains de gérer de nombreuses situations conflictuelles ou nécessitant des réactions rapides. Par exemple, le chant des oiseaux permet à ceux-ci de se repérer dans la forêt, de s'informer sur la présence d'un prédateur, etc (cf. [McFarland, 1987][Noble, 1999]). Les hommes utilisent aussi toutes sortes de signaux sonores (klaxons, cris, sirènes) ou lumineux (feux de signalisation, gyrophares, lampes) pour se faire comprendre dans des situations soumises à des règles (croisements de routes, circulation) ou libres (déplacements dans une foule, dans un embouteillage, dans un couloir, etc...).

Il est intéressant de reprendre ces notions de signaux, de diffusions et de priorités pour les appliquer à des robots qui doivent réagir rapidement et s'adapter aux situations d'environnements dynamiques.

Ainsi, nous proposons de traiter les problèmes de conflits et de blocages entre agents en n'utilisant que des perceptions et des signaux locaux. Nous montrons que ces choix, de faibles coûts et de faibles complexités, peuvent être suffisants pour détecter et résoudre des blocages spatiaux. L'idée générale étant de propager, de proche en proche, des informations entre les agents sous forme de signaux, qui les mènent à des comportements collectifs intelligents.

Le module de sélection d'action proposé au chapitre 7 prévoit qu'une forte insatisfaction due à une gêne spatiale déclenche l'émission d'un signal de répulsion. Si un blocage a lieu entre deux agents, celui qui émet l'insatisfaction "la plus forte" parvient à repousser l'autre. En effet, celui qui *perçoit un signal répulsif plus fort que le sien bascule en état d'altruisme* et exécute le déplacement adéquat.

**Principe de résolution :** Lorsqu'ils agissent, les agents émettent des signaux de satisfaction interactive pour signifier à leurs voisins immédiats des gênes ou des blocages éventuels. La résolution des blocages est basée sur le fait que ce sont les agents les moins contraints qui doivent adopter un comportement altruiste (ici de recul) devant les autres. Ainsi, les agents les plus contraints (les plus insatisfaits) propagent des vagues d'insatisfaction autour d'eux jusqu'à faire s'écarter ou reculer les gêneurs.

S'il y a *plusieurs* agents bloqués en un point de passage trop étroit, la seule possibilité de résolution, c'est que l'ensemble des influences détermine un sens unique "d'écoulement", permettant la libération des agents. De façon évidente, un écoulement ne peut se produire que d'une zone bloquée/saturée vers une zone libre.

## 9.4 Eco-robots et règles comportementales

Le mécanisme de résolution proposé s'appuie directement sur le fonctionnement générique du modèle Satisfaction-Altruisme. Il s'agit ici de préciser le calcul des valeurs émises et relayées par les agents, dans le but d'établir des vagues d'insatisfactions des agents les plus contraints vers les moins contraints.

En section 6.3 nous avons établi un principe de priorité des signaux le plus intenses (en valeur absolue) sur les autres. Pour assurer la propagation des *répulsions*, il est donc nécessaire de compléter ce critère par une priorité des signaux répulsifs sur les signaux attractifs.

**Principe de priorité aux répulsions :** Les agents sélectionnent et diffusent en priorité les signaux répulsifs ; entre deux signaux de même signe la priorité est au plus intense.

Justification : Les signaux répulsifs sont utilisés pour résoudre les conflits spatiaux entre agents. S'ils ne sont pas prioritaires, de fortes attractions peuvent empêcher leurs diffusions et donc maintenir les conflits. De plus, entre plusieurs signaux répulsifs, il faut que la priorité soit au plus intense, puisque cette intensité est une mesure du niveau de contrainte/blocage de l'agent. Il s'agit, contrairement à l'approche de [Vaughan et al., 2000], de prendre en compte la perception locale de l'agent *et celle de ses voisins*, de telle sorte que les interactions ne soient pas limitées à des agressions "face à face".

Conséquence : Pour réaliser toute fonction dans l'environnement (et en particulier de survie) un agent doit tout d'abord conserver la capacité de se déplacer (cf. chapitre 3). Ainsi, la réaction altruiste aux signaux répulsifs, assurant le déplacement des agents, doit être appliquée comme une fonction de survie prioritaire sur toutes les autres.

Nous avons présenté au chapitre 7 la formation implicite de chaînes répulsives, qui est une répétition du processus :

[ agent A : perception d'un gêneur B  $\Rightarrow$  émission d'une répulsion  
 [ agent B : réception du signal de A, altruisme  $\Rightarrow$  essayer de fuir

Cependant, la répétition de ce processus peut être relativement lente : lorsque l'agent B tente de fuir, si un agent C le gêne, il ne va émettre une répulsion que lorsque sa satisfaction personnelle aura fortement chuté.

Pour accélérer ce processus de propagation, nous avons introduit, comme pour les signaux attractifs, un processus de relais immédiat des signaux perçus (cf. détail section

suivante). Ce qui s'exprime par le schéma suivant :

$$\begin{array}{l} \text{[ agent A : perception d'un gêneur B } \Rightarrow \text{ émission d'une répulsion } I_{A/B} \\ \text{[ agent B : réception du signal } I_{A/B}, \text{ altruisme } \Rightarrow \left\{ \begin{array}{l} \text{essayer de fuir} \\ \text{si perception gêneur : emettre } I_{A/B} \end{array} \right. \end{array}$$

Le principe de priorité aux répulsions énoncé précédemment permet de généraliser ce processus à un ensemble d'agents évoluant à proximité.

### 9.4.1 Une application de l'éco-résolution à des robots mobiles

L'expression de la propagation des insatisfactions que nous venons de donner s'apparente à un processus d'éco-résolution [Ferber, 1989] [Bura et al., 1991] [Drogoul, 1993] (introduit en section 3.2.2). En effet, nous pouvons exprimer le principe de résolution des blocages spatiaux entre robots par l'écriture suivante :

#### Structure des éco-robots

Chaque agent ne peut être que dans l'un des trois états suivants :

- satisfait = robot en mouvement libre ( $P > 0$ ),
- en recherche de satisfaction = robot bloqué/gêné par des voisins ( $P < 0$ ),
- en recherche de fuite = état d'altruisme lors de la perception d'un signal  $I_{e(<0)} < P$ .

#### Comportements des éco-robots

Comme dans l'éco-résolution, les agents ont pour but d'être satisfaits (ou d'éviter/résoudre leur insatisfaction)

- tout robot bloqué/gêné par un voisin l'agresse par l'émission d'un signal répulsif, la contrainte est codée par l'intensité de l'insatisfaction,
- la réception d'un signal dont la valeur représente une insatisfaction supérieure à celle du robot le fait passer en état d'altruisme, donc l'oblige à fuir.
- L'état de fuite/altruisme provoque, en présence d'un gêneur, l'émission d'un signal répulsif de même intensité que celui qui est perçu : c'est le transfert de la contrainte.

Cette écriture du comportement des robots en éco-agents n'est toutefois pas suffisante pour garantir une résolution de leur conflits spatiaux. En effet, il est nécessaire d'y intégrer la dimension spatiale (ou de mobilité) du problème. Nous revenons en fin de chapitre sur la comparaison entre l'éco-résolution et le modèle que nous proposons.

Le niveau de satisfaction des agents est directement lié à leur capacité de se mouvoir : agent en mouvement  $P \nearrow$ , agent bloqué  $P \searrow$ .

Supposons un ensemble d'agents paralysés par des conflits spatiaux. Certains agents déclenchent alors des répulsions qui, par propagation, initient un processus de déblocage. Cependant, les premiers mouvements réalisés induisent aussi une remontée des niveaux de satisfactions. Les agents, à nouveau satisfaits, stoppent alors leurs émissions de répulsions, annulant ainsi le début de la résolution. Le système retrouve une situation de blocage.

Ce processus se réitère et soit le système oscille indéfiniment entre mouvement et paralysie, soit au mieux il progresse lentement vers une résolution.

Ce problème nous a mené à compléter le modèle d'éco-robots par les notions de *persistance* et d'*amplification* des signaux émis.

### 9.4.2 Règles de résolution

Nous exprimons le modèle satisfaction-altruisme sous formes de règles comportementales locales, en reprenant le modèle initial et en le complétant d'une définition plus fine des signaux répulsifs émis. Le coefficient d'altruisme  $\alpha$  a pour valeur 0.5, ainsi il n'apparaît pas dans les règles données ci-dessous (nous justifions ce choix plus loin).

**Règle 1 (de passage en état d'altruisme).** Si l'agent n'est pas en train d'émettre, toute perception d'un signal prioritaire sur sa propre satisfaction personnelle  $P$  déclenche son passage en état d'altruisme ; si l'agent émet il faut aussi que le signal perçu soit prioritaire sur le sien.

Justification : Tout agent recevant un signal exprimant une insatisfaction plus grande que la sienne bascule en état d'altruisme (test d'altruisme chapitre 6). Cette condition garantit qu'entre deux agents, celui qui passe en état d'altruisme est le moins insatisfait des deux. La deuxième condition est justifiée à la règle 3.

**Règle 2 (de déclenchement d'une répulsion).** Un agent qui n'est pas en état d'altruisme déclenche l'émission d'une répulsion lorsqu'il dépasse un seuil d'insatisfaction et qu'il perçoit la présence d'agents voisins gênants, alors  $I(t) = P(t)$ .

Justification : Cette règle assure l'émission d'une répulsion lorsqu'un agent est gêné/bloqué par un autre ( $P < \mu \leq 0$ ). La valeur du signal émis est égale à l'insatisfaction de l'agent  $P(t)$ , car elle représente l'intensité des contraintes subies.

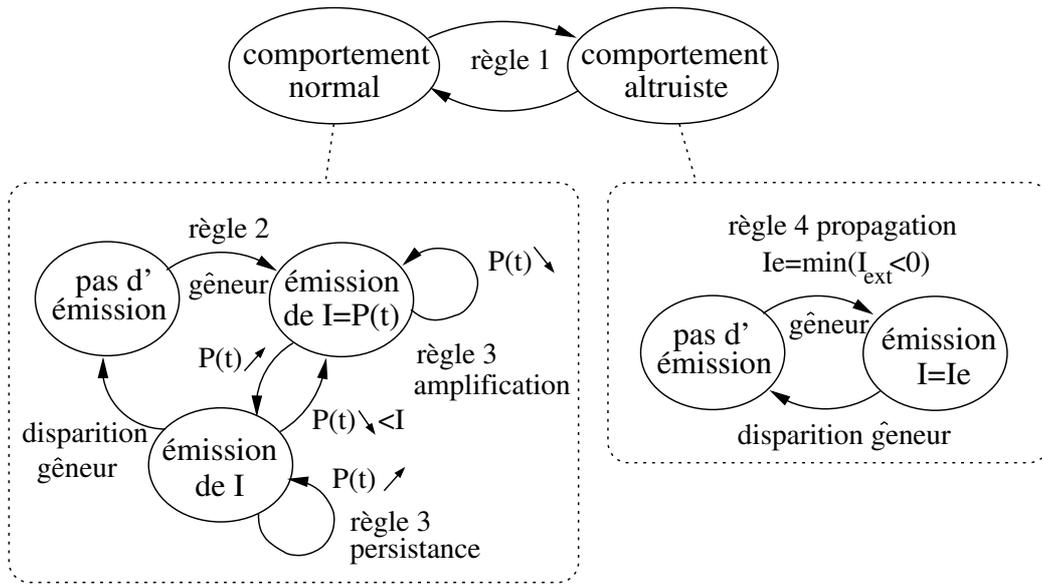


FIG. 9.2 – Graphe d'états finis du comportement et des signaux émis par un agent.

**Règle 3 (de persistance et d'amplification).** Lorsqu'un agent a déclenché l'émission d'une répulsion, il continue à émettre cette répulsion tant qu'il perçoit un agent dans sa direction. La valeur émise reste constante ou s'amplifie lorsque l'insatisfaction de l'agent augmente :  $P(t) \searrow < I(t) \Rightarrow I(t) = P(t)$  sinon  $I(t) = I(t-1)$ .

Justification : Cette règle est nécessaire pour qu'un agent persiste dans sa tentative de repousser un gêneur. En effet, lorsqu'un agent commence à repousser un gêneur sa satisfaction personnelle remonte vers des valeurs positives. Or, par application de la règle 2, il risque d'arrêter son émission avant de s'être libéré du gêneur, ce qui mènerait inévitablement à un processus oscillatoire {blocage, déblocage}.

Ainsi, l'intensité du signal émis ne doit pas décroître lors de la remontée de  $P$ , alors que l'agent n'est pas encore libéré ( $I(t) = I(t-1)$ ). Cette même raison explique la présence du deuxième test dans la première règle, qui consiste à déclencher l'altruisme que si le signal extérieur est plus fort que celui émis par l'agent.

Si l'agent ne parvient pas à se libérer rapidement (ex. arrivée de nouveaux agents insatisfaits) son insatisfaction augmente  $P(t) \searrow$ , alors l'intensité du signal qu'il émet doit s'amplifier aussi :  $I(t) = P(t)$ .

**Règle 4 (de propagation).** Soit  $I_e$  le signal répulsif *prioritaire* reçu par un agent en état d'altruisme, alors s'il perçoit un agent dans sa direction il émet un signal égal à  $I_e$ .

Justification : Cette règle est nécessaire à la propagation des insatisfactions entre agents. Elle permet d'assurer le transfert des signaux les plus prioritaires, c'est-à-dire des plus fortes insatisfactions, des agents les plus contraints aux moins contraints.

Ces quatre règles s'écrivent formellement sous forme d'un algorithme :

- 
1. [r1] Si  $I_e < (0, P, I)$  alors { altruisme=vrai ;
  2. [r4] Si gêneur=vrai alors  $I = I_e$  sinon  $I = 0$  }
  3. Sinon { altruisme=faux
  4. Si gêneur=vrai alors
  5. [r2] { Si  $I = 0$  et  $P < \mu$  alors  $I = P$
  6. [r3] Si  $P < I$  alors  $I = P$
  7. } sinon  $I = 0$
  8. }
  9. Si  $I \neq 0$  alors émettre  $I$
- 

La figure 9.2 donne sous forme d'un graphe d'états finis une représentation plus lisible de l'algorithme et de son fonctionnement.

## PROCESSUS DE RÉOLUTION

### 1. Gênes

Considérons un ensemble d'agents en situation de blocages/conflicts d'accès (aucun n'émet de signaux). Les agents dont la satisfaction personnelle chute au dessous du seuil  $\mu$  déclenchent l'émission d'un signal répulsif égal à leur insatisfaction :  $I = P$  (règle 2, lignes 4 et 5). A ce stade tous les agents perçoivent au moins un signal répulsif.

Chaque agent évalue tout d'abord  $I_e$  la valeur du signal le plus prioritaire qu'il perçoit, il exécute alors la règle 1 qui le mène

- soit dans l'état d'altruisme car le signal  $I_e$  est prioritaire sur son niveau de satisfaction ou sur son éventuel propre signal,
- soit à conserver son état normal.

S'il n'y a qu'un signal émis dans le système, un seul agent reste dans l'état normal, tous les autres basculent en altruisme (vérifiant  $I_e < P$ ). Si plusieurs agents émettent des répulsions, seulement ceux qui diffusent un signal prioritaire sur les autres restent dans l'état normal (vérifiant  $I \leq I_e$ ).

## 2. Amplification

Examinons le comportement des agents pour chacun des deux états possibles :

1. le ou les agents qui sont restés dans l'état normal :
  - s'ils ne parviennent pas à se déplacer leur insatisfaction grandit, alors la valeur qu'ils émettent s'amplifie (règle 3)
  - s'ils parviennent à se déplacer, leur satisfaction remonte, mais ils continuent à émettre un signal de valeur constante jusqu'à s'échapper (persistance règle 3)
2. les agents en état d'altruisme tentent de s'éloigner et relayent, s'ils sont gênés, le signal prioritaire qu'ils reçoivent (règle 4).

## 3. Résolution

- cas 1 : si tous les agents en état d'altruisme peuvent se déplacer, c'est-à-dire s'éloigner, alors les agents les plus insatisfaits se libèrent.
- cas 2 : si un ou plusieurs agents en état d'altruisme ne peuvent pas se déplacer, alors leur insatisfaction grandie. L'amplification de ces insatisfactions les mènent à l'une des deux situations suivantes :
  - Soit l'agent en état d'altruisme parvient à repousser ses propres gêneurs et il libère celui qui émet la répulsion.
  - Soit l'agent en état d'altruisme atteint un niveau d'insatisfaction plus important que celui qu'il perçoit. Dans ce cas, il sort de l'état d'altruisme (règle 1) et il émet à son tour une répulsion. Cette répulsion étant nécessairement plus forte que la précédente, sa probabilité de se libérer est alors plus grande, le processus général se répète.

**Conséquence** : La résolution d'un blocage nécessite en premier lieu une bonne définition de la mesure d'insatisfaction. Elle doit évoluer suivant le niveau de paralysie/contrainte d'un agent, en fonction de sa situation locale : types et nombres d'obstacles (cf. applications). De plus, la configuration géométrique de l'environnement et des agents détermine l'existence d'une solution (cf. fin applications).

La méthode proposée se base sur la propagation des insatisfactions des agents : si  $P < 0$  alors  $I = P$ . L'objectif étant que les plus insatisfaits repoussent les autres. Ce comportement ne peut être assuré que si le test d'altruisme est effectué par comparaison directe entre  $P$  et  $I_e$ , c'est-à-dire en prenant  $\alpha = 0.5$ .

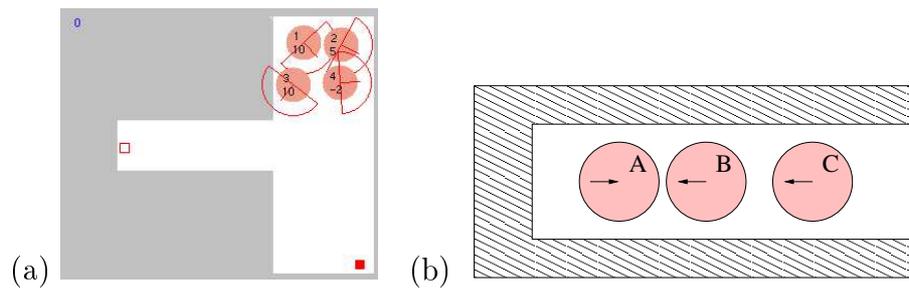


FIG. 9.3 – (a) Etat initial d’une simulation pour le problème du couloir avec 4 robots  
(b) schéma d’une situation conflictuelle.

## 9.5 Applications

### 9.5.1 Piéger les robots

Nous avons montré dans les précédents chapitres que l’architecture intégrait en son coeur les notions d’évitement d’obstacles et de coordination des trajectoires. Ces capacités sont suffisantes pour assurer aux agents des déplacements fluides parmi des obstacles fixes ou mobiles. Dans ce chapitre nous nous concentrons sur les situations de blocages/conflicts qui surviennent dans des zones réduites de l’environnement.

Notre démarche est d’étudier la situation locale la plus critique afin d’établir une solution robuste. Ainsi, nous traitons en détails le cas de l’utilisation simultanée, par des agents indépendants, d’un couloir fermé (une impasse) où il leur est impossible de se croiser. C’est une situation piège pour des robots mobiles autonomes puisqu’ils n’ont qu’une perception locale et non mémorisable de leur environnement [Haninayan and Lumelsky, 1994].

Formellement le problème peut s’énoncer ainsi :

- 
- Soit un ensemble homogène ou hétérogène d’agents exécutant des tâches indépendantes et nécessitant des déplacements,
  - Soit un étroit couloir en impasse que doivent emprunter dans sa totalité tous les agents,

Problème : comment garantir que tous les agents puissent réaliser leur tâche dans le couloir en un temps fini ?

---

**Créer des conflits** Considérons un système de  $N$  robots autonomes situés dans une pièce dont la seule sortie soit un couloir sans issue (cf. fig. 9.3.a). Supposons que les agents est pour mission de sortir de cet environnement et qu’ils utilisent comme stratégie la recherche

aléatoire et l'exploration des passages.

Le comportement d'un agent autonome va nécessairement être au bout d'un certain temps : détection du couloir, exploration et finalement découverte d'une impasse et non d'une sortie. Dans ce cas l'agent déclenche une tâche d'éloignement de l'impasse jusqu'à en sortir.

Supposons que cette "expérience" soit menée par plusieurs agents simultanément, alors rapidement des conflits vont apparaître, puisque certains agents vont vouloir sortir du couloir tandis que d'autres vont essayer d'y entrer. Rappelons que le couloir est trop étroit pour permettre tout croisement d'agents. Une telle situation de conflits est schématisée en figure 9.3.b.

Nous avons défini au chapitre 4 la satisfaction par une mesure incrémentale de la progression des actions (formule 4.2). Notre méthode de résolution nécessite seulement de spécifier plus précisément le calcul de l'insatisfaction des agents lorsqu'ils sont immobilisés.

### Pénalisation d'un blocage

Nous distinguons deux type d'obstacles : les objets (supposés non déplaçables) et les agents. En effet, les agents sont des obstacles particuliers car ils ont la capacité de se déplacer et en particulier de répondre à des signaux d'influences. Par conséquent, ils représentent des contraintes environnementales moins fortes que des obstacles fixes.

La pénalisation de la satisfaction dépend donc du type d'obstacles qui gêne l'agent. Comme cette immobilisation peut être due à plusieurs agents et plusieurs objets à la fois, nous avons défini un calcul de cette pénalisation suivant les différents capteurs de l'agent.

**Hypothèse** : chaque agent possède une perception locale et omnidirectionnelle de son environnement, suivant une discrétisation donnée par le nombre de capteurs qu'il détient (ou du nombre de zones d'analyses pour une "vision" par caméra).

---

la pénalisation d'un blocage vaut à l'instant  $t$  :  $v = N_1.\theta + N_2.\theta'$

$N_1$  est le nombre de capteurs vérifiant la présence d'un agent,

$N_2$  est le nombre de capteurs vérifiant la présence d'un obstacle,

$\theta$  est la valeur de pénalisation pour un blocage dû à *un agent* ( $\theta < 0$ ),

$\theta'$  est la valeur de pénalisation pour un blocage dû à *un objet*, tel que  $\theta' < \theta$ .

---

Les agents sont de forme cylindre (de rayon  $R$ ) et ont les caractéristiques suivantes :

- Perception de l'environnement dans un disque de rayon  $3.R$ , le calcul de la trajectoire est déterminé par les capteurs du secteur  $[-\frac{\pi}{2}, \frac{\pi}{2}]$  orienté dans la direction de déplacement.
- Communication par diffusion de signaux omnidirectionnels et de portée  $3.R$ . Un tel rayon limite la réception à au plus un agent par direction.

### 9.5.2 Étude, Simulations et analyses

#### Représentation des conflits, des satisfactions et des signaux

Nous utilisons deux types de schémas, qui sont des projections des positions et des états des agents dans le plan :

- schéma des *satisfactions personnelles* : cette représentation permet de donner à un instant  $t$  les positions des agents dans le corridor ainsi que leur niveau de satisfaction personnelle ( $P$ ), ceci par une projection dans le plan  $\langle x, P \rangle$ . La direction recherchée par chaque agent est décrite par une flèche horizontale à la base de chaque agent. Ainsi, les conflits d'espaces (de trajectoires) sont directement lisibles. Le fond du corridor se trouve en  $x = 0$  (cf. exemple fig. 9.5).

- schéma des *signaux d'insatisfactions* : cette représentation projette les agents dans le plan  $\langle x, -I \rangle$ . Elle permet de représenter les signaux émis par les agents ainsi que leurs effets. Un agent en état d'altruisme est noté avec la lettre A (cf. exemple fig. 9.4).

#### Gradient d'insatisfactions - ou considérations d'implémentations réelles -

Un agent propage un signal répulsif en émettant une valeur égale à celle qu'il reçoit. Cette opération, logique en théorie, comporte des défauts dans le cadre d'une implémentation réelle.

En particulier, toute coupure temporaire de transmission dans une chaîne de répulsion peut provoquer le changement de direction d'un agent, sans que celui-ci retrouve la "bonne direction" dès la coupure terminée. La figure 9.4.A détaille le déroulement d'un tel processus néfaste. Nous avons éliminé ce type de problème en introduisant une information implicite dans l'opération de propagation.

La solution proposée consiste à relayer les signaux de façon *amortie*. Le mécanisme est simple : un agent en état d'altruisme, percevant un signal  $I_e < 0$ , émet un signal d'intensité légèrement moins forte :  $I_e + \epsilon$  ( $\epsilon > 0$ ). Ainsi, le front se propage mais en s'amortissant (en fonction de la distance à la source), créant ainsi un gradient représentant le "sens d'écoulement" des agents.

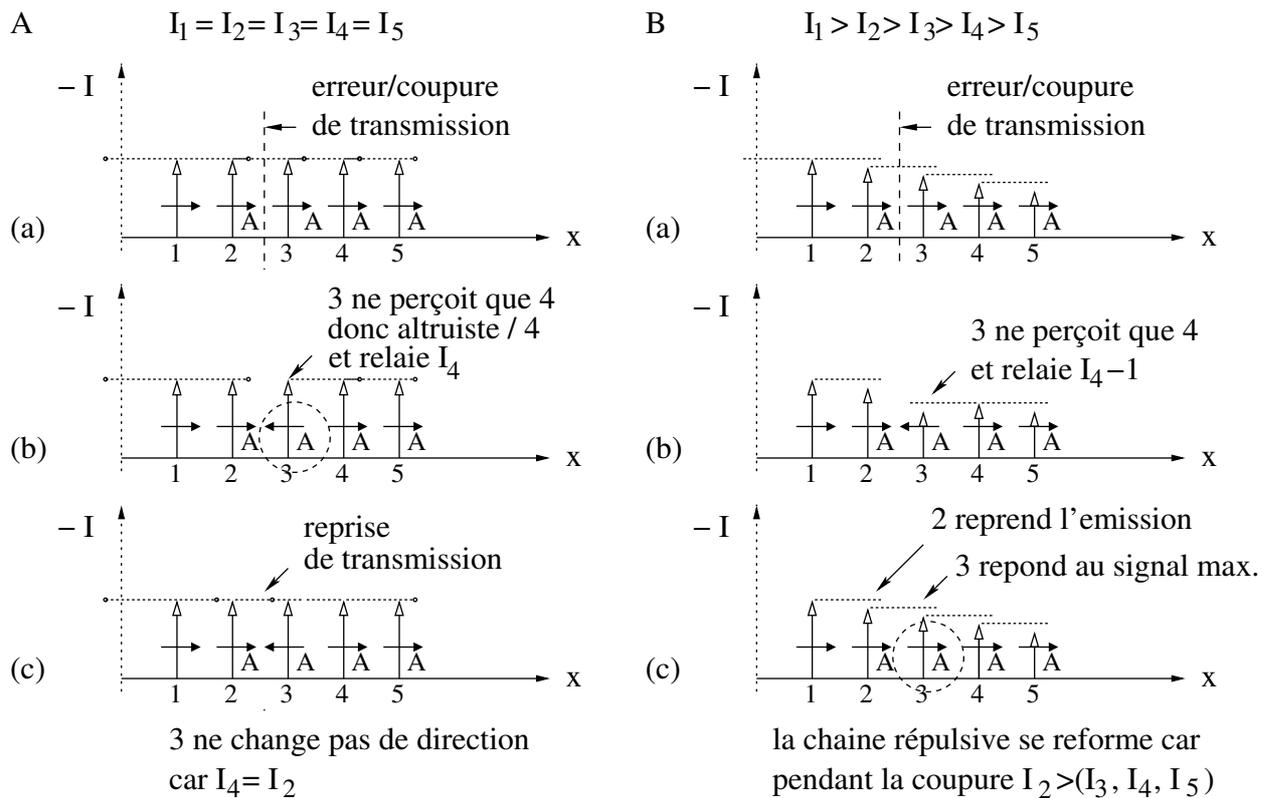


FIG. 9.4 – (A) propagation à valeur égale : incohérence de la chaîne répulsive en cas de coupure (B) propagation amortie : résolution du problème (gradient d’insatisfaction).

Dans le cas du problème de “coupure” d’une chaîne répulsive, ce gradient permet aux agents de retrouver très vite le bon sens de déplacement (cf. détails solution fig. 9.4.B).

### Résolution des conflits dans le corridor

La figure 9.5 donne une description schématisée du processus de résolution. La preuve de la résolution systématique des conflits d’accès dans un corridor fermé est donnée en annexe 14.2.1. Notons que ce type de preuve ne peut être envisagée que pour des environnements spécifiques (ou des “classes” d’environnements). Le problème d’une preuve de complétude reste ouvert. Notons par ailleurs que le problème général de la *planification* des trajectoires de plusieurs disques dans un milieu rempli d’obstacles polygonaux est NP-difficile au sens fort [Spirakis and Yap, 1984], la complexité est de l’ordre du nombre de côté des polygones mis à la puissance du nombre de disques.

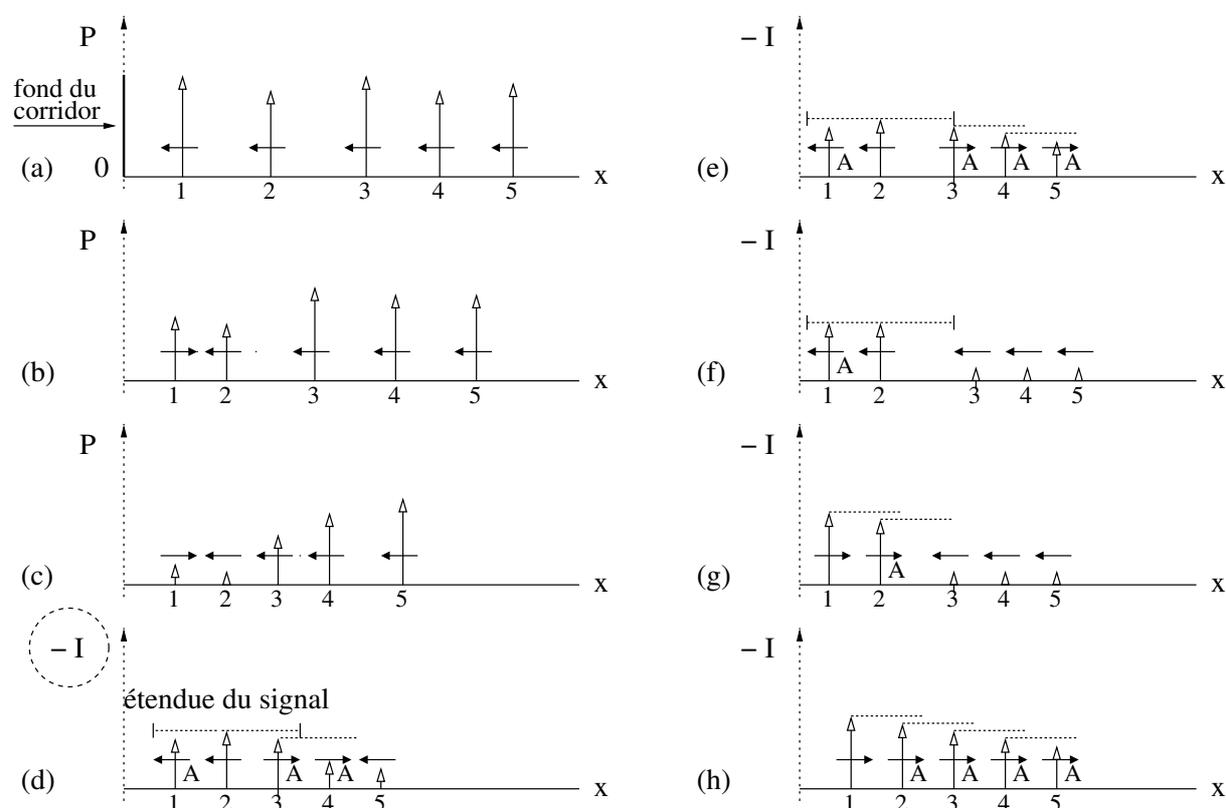


FIG. 9.5 – Évolution d’une situation d’exploration du corridor par plusieurs agents

- (a) tous les agents explorent le corridor, les satisfactions  $P$  sont très élevées.
- (b) l’agent 1 a découvert que le corridor est fermé, il décide d’en sortir, et se retrouve alors en conflit avec l’agent 2, leur satisfaction  $P$  chute. Les autres agents continuent leur progression.
- (c) Les agents 3 et 4 “butent” sur le blocage. Les satisfactions  $P$  continuent à décroître.
- (d) L’agent 2, le plus insatisfait, émet une répulsion plus forte que celles de 1 et 3. Ces agents passent en état d’altruisme : 1 part en direction du fond du corridor, l’agent 3 vers la sortie. La répulsion se propage aux agents 3 et 4 (état d’altruisme).
- (e) L’agent 1 bute sur le fond du corridor. La répulsion de 2 s’est propagée à 3, 4 et 5.
- (f) L’agent 1 est bloqué, son insatisfaction augmente. L’agent 3 est sorti du champ de répulsion de 2, les agents 3, 4 et 5 repartent en exploration.
- (g) L’insatisfaction de l’agent 1 est plus forte que le signal répulsif perçu, il émet alors une répulsion. Cette répulsion dépasse celle de l’agent 2 qui bascule en altruisme.
- (h) Le front de répulsion ainsi créé se propage aux autres agents, ils prennent tous la direction de la sortie. La persistance de l’émission de l’agent 1 lui permet de s’échapper du corridor.

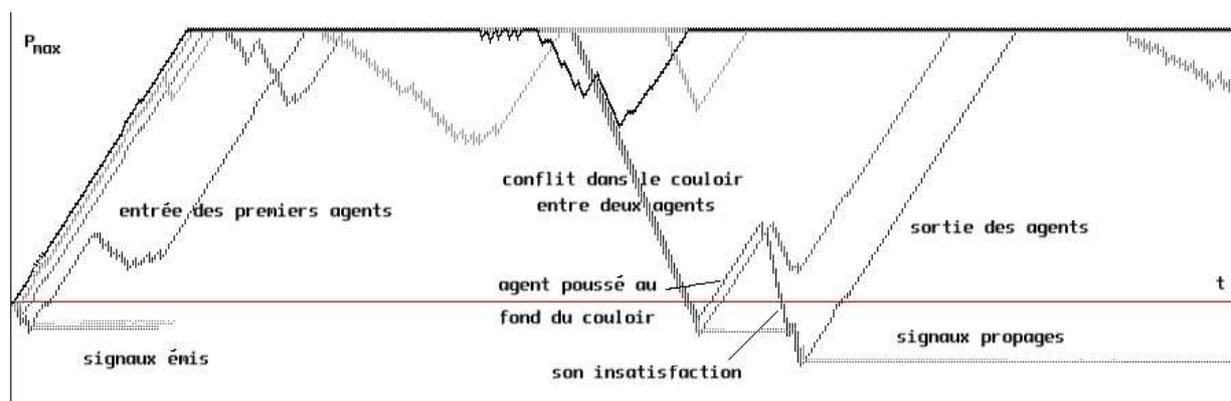


FIG. 9.6 – Tracés d'hédographes superposés pour 5 agents dans une simulation de résolution du problème de l'impasse.

### Simulations et résultats

L'outil de simulation présenté au chapitre 8 a été adapté pour donner une représentation plus proche de la réalité des systèmes étudiés (la taille des agents et leur zone de perception sont paramétrables).

Chaque agent est de forme cylindrique (rayon  $r_a$ ), sa trajectoire est déterminée par sa perception du secteur  $[-90^\circ, 90^\circ]$  devant lui et sur une portée de rayon  $2.r_a$ . Les signaux émis, à valeurs dans  $[-127, 127]$  sont diffusés dans un disque de rayon  $3.r_a$  (assurant des communications uniquement entre agents très proches). L'implémentation du calcul de trajectoire (eq. 6.2) est donné en annexe 14.1.2.

Si deux agents s'échangent des signaux répulsifs de valeurs égales, alors suivant une probabilité de 0.5, leur niveau de satisfaction personnelle subit une légère remontée. Ceci dans le but de rompre l'équilibre et de créer une réaction si la situation reste figée.

#### A. Corridor fermé

Dans les simulations menées, tous les agents cherchent à réaliser des déplacements en contournant les obstacles fixes et les autres robots. Toutefois, dans un couloir, un robot *insiste pour avancer face à un autre*, il ne recule que s'il est en état d'altruisme.

La résolution obtenue figure 9.7 montre que le comportement des agents correspond au processus théorique présenté à la section précédente (une description plus détaillée de cette simulation est donnée dans [Simonin and Ferber, 2001]). Ces simulations ont été menées avec un nombre d'agents variant de 2 à 8, la résolution est systématique, et ce quelle que soit la longueur du corridor.

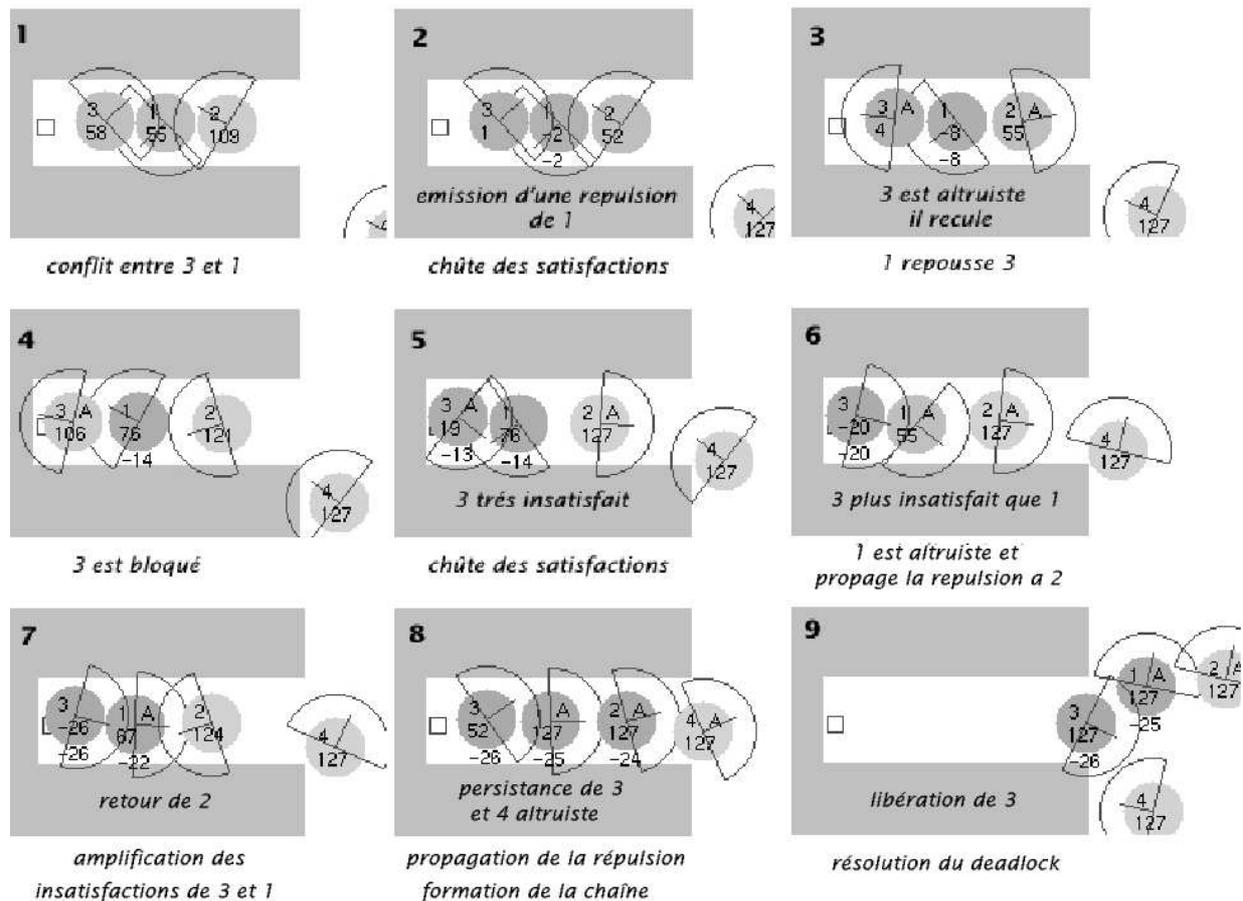


FIG. 9.7 – Simulation : résolution du problème de l'impasse avec 4 agents indépendants.

Toutes les simulations ont été accompagnées, pendant leur exécution, du tracé des valeurs de satisfactions personnelles (ou hédographes), des états d'altruisme et des signaux émis. Nous donnons en figure 9.6 un exemple d'un tel tracé pour une résolution du problème du couloir avec 5 robots. Nous n'avons affiché ici que les valeurs des satisfactions personnelles et des signaux émis.

Ce résultat "statique" est difficile à lire, mais la vision de ces tracés pendant l'exécution des simulations permet de suivre très clairement l'évolution des états des agents et de leurs émissions.

## B. Couloirs, pièces et impasses

Nous avons évalué le comportement des agents dans un environnement très contraint composé de pièces, de couloirs, d'impasses et d'attracteurs (il reproduit une grande partie

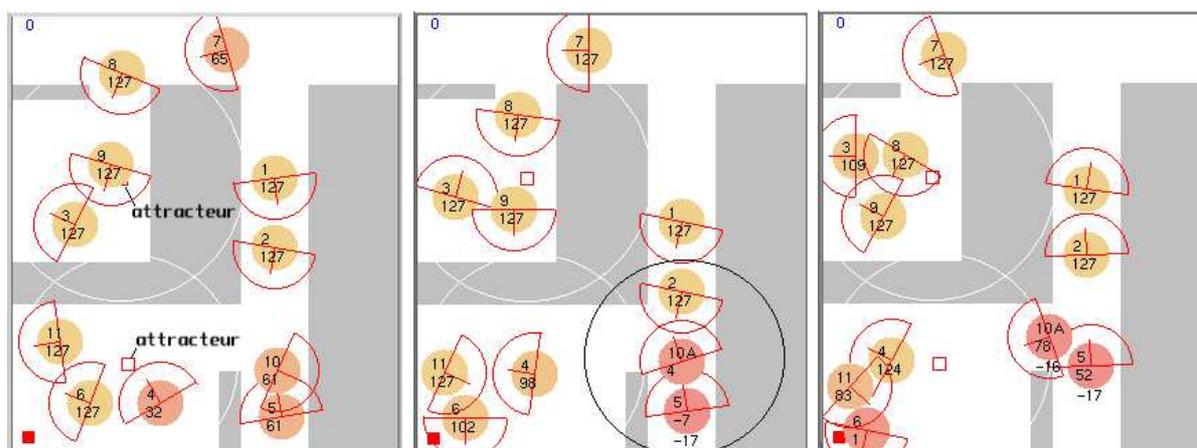


FIG. 9.8 – Évolution de 11 agents “explorateurs” dans un environnement contraint (type [Vaughan et al., 2000])

de l’environnement test proposé dans [Vaughan et al., 2000]). Pour provoquer des conflits, le comportement de chaque agent est défini ainsi :

- Si l’agent perçoit un attracteur (carré vide fig. 9.8) il doit l’atteindre, alors il s’en désintéresse jusqu’à ce qu’il quitte la pièce,
- Sinon l’agent explore aléatoirement l’environnement mais insiste pour poursuivre sa route lorsqu’il rencontre un autre agent.

Les simulations ont été effectuées volontairement avec de nombreux agents, nous avons alors observé l’apparition inévitable de conflits et leur résolution.

- Des conflits se produisent autour des attracteurs (en particulier en début de simulation) : le modèle permet aux agents d’atteindre à tour de rôle la source, puis de s’en s’écarter pour laisser la place aux autres. Ce résultat est à la fois fluide (composition vectorielle des perceptions d’obstacles et des signaux) et auto-organisé suivant l’état de blocage des agents (qu’on pourrait qualifier d’impatience ici).
- Tout “groupe” d’agents en conflit dans une impasse en sort par l’exécution du processus décrit plus haut.
- Lorsque deux chaînes répulsives se rencontrent face à face dans un couloir, soit l’une des deux repousse immédiatement l’autre (agrandissant la chaîne), soit les signaux s’équilibrent au point de rencontre (cette situation peut exister par l’amortissement des signaux). Or, dans ce cas, les agents situés à la rencontre des deux chaînes

deviennent plus vite insatisfaits que ceux des extrémités, ils parviennent alors à les repousser.

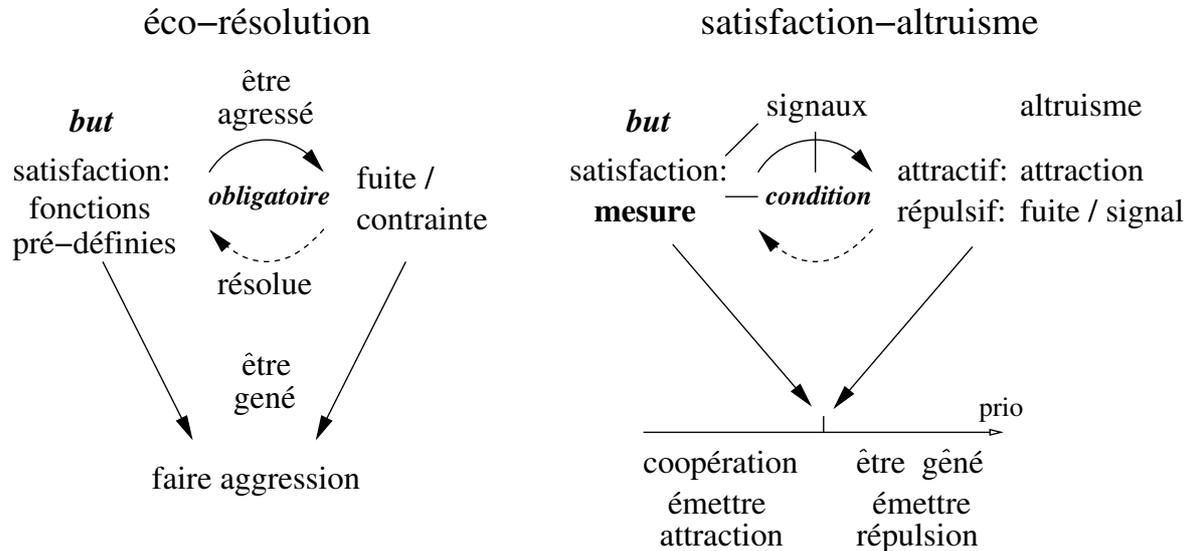
- Lors du mouvement des chaînes répulsives, nous avons observé un résultat systématique très intéressant : sous la pression des signaux et des d'obstacles, les agents exploitent naturellement les passages/ouvertures perpendiculaires aux directions de répulsions, cf. fig. 9.8 agent 10 (comportement que nous qualifions d'émergeant dans la mesure où nous ne l'avons pas prévu explicitement dans le modèle). Ceci permet de limiter le déplacement des agents repoussés et de libérer plus vite l'agent contraint.
- Les situations de tentatives d'accès simultanés d'un passage par deux agents "côte à côte" (sortie d'une pièce, angle de deux couloirs) provoquent des blocages. Ils sont immédiatement résolus dès lors que l'un des agents est plus insatisfait que l'autre.
- Enfin, le simulateur permettant d'inhiber temporairement certains aspects du modèle, on constate qu'en l'absence d'émission de signaux, les agents se bloquent très vite dans des situations insolubles. Avec émission de signaux, mais sans propagation, certains de ces conflits surviennent et ne trouvent pas de solutions. L'émission de signaux et leur propagation sont donc nécessaires à la résolution de ces problèmes.

### **Remarques sur l'existence d'une solution :**

- Si nous considérons de vastes pièces reliées par des couloirs vides, les agents peuvent toujours repousser leurs voisins jusqu'à atteindre la pièce voulue (hypothèse d'étude de [Vaughan et al., 2000]).

- Dans le cas contraire, rien ne garantit qu'une solution puisse être trouvée. Il existe des environnements très contraints pour lesquels le déplacement possible des agents vers leur but nécessite une longue réflexion (humaine). Nous n'avons pas testé le modèle sur ce type d'environnements extrêmes, généralement problématiques pour des systèmes réactifs. Toutefois, le modèle a montré une tendance naturelle à bien exploiter les espaces offerts par l'environnement lorsque les agents se repoussent. Ceci nous laisse penser que de nombreux cas extrêmes peuvent être résolus, en un temps fini, par notre approche.

## 9.6 Eco-résolution et modèle satisfaction-altruisme



L'éco-résolution est une méthode de traitement distribuée des problèmes très générique et puissante. Dans le cadre du fonctionnement d'agents situés, nous comparons le comportement des éco-agents avec celui défini par le modèle satisfaction-altruisme.

- Dans un processus d'éco-résolution, la satisfaction des agents est souvent définie par un ensemble de fonctions liées à la décomposition des situations potentiellement rencontrables par les agents. En cas d'insatisfaction (gêne) l'agression passée aux voisins est une requête non nécessairement homogène avec la définition de la satisfaction. Au contraire, dans notre modèle, nous nous sommes attaché à créer cette équivalence pour établir une *condition* à la réaction des voisins.

- Ainsi nous avons reformulé la satisfaction sous forme d'une mesure continue liée aux perceptions courantes de l'agent (chapitre 4). Cette mesure, adaptée aux actions d'un agent situé (simulé ou réel) nous a permis d'établir une condition d'interaction entre les agents (le test d'altruisme, chapitre 5). Celui-ci permet de décider si une influence extérieure est suffisamment forte pour qu'un agent s'implique dans une interaction avec un voisin. Ce choix est particulièrement intéressant dans le cadre de tâches coopératives (chapitre 8), mais également pour la résolution des conflits spatiaux (notions de priorités, d'amplification et de persistance des insatisfactions).

- Le modèle satisfaction-altruisme introduit la notion de coopération, par signaux d'attractions, dans la boucle d'interaction entre les agents. Là encore, l'information transmise est homogène avec l'évaluation de la satisfaction. Par contre, nous avons été amené à poser

une priorité aux répulsions pour assurer la résolution des conflits spatiaux. Il faut donc voir cette capacité d'attraction comme un complément qui n'interfère pas avec le principe d'éco-résolution, mais y apporte une forme de coopération pouvant accélérer le processus de résolution.

- Contrairement à l'éco-résolution, le modèle satisfaction-altruisme limite les contraintes passées à de simples signaux. Toutefois, cette spécialisation, d'une sémantique certes limitée, apporte en contrepartie les qualités d'un processus d'interaction en temps réel. Ce chapitre a montré tout l'intérêt d'une telle démarche pour la navigation d'agents autonomes (le chapitre 12 présente son implémentation sur des robots réels).

## 9.7 Bilan

Ce chapitre a permis de montrer qu'en affinant la définition des signaux répulsifs, le modèle satisfaction-altruisme était suffisant pour gérer la résolution de certains conflits spatiaux, et mieux, d'obtenir de véritables comportements collectifs cohérents. Nous avons montré que la formation de chaînes répulsives est un processus d'éco-résolution, mais qu'il devait être complété des notions de persistance et d'amplification pour s'adapter à des agents situés.

Le modèle proposé ne garantit pas une résolution de tous les conflits spatiaux envisageables. Nous avons simplement établi un résultat sur un espace critique, celui des passages en impasses. Le modèle s'est tout de même montré pertinent, en simulation, pour la navigation de nombreux agents dans des espaces relativement contraints.

Plus généralement, nous avons constaté que ces systèmes d'agents situés sont très complexes à appréhender formellement de par leur dynamique. Dans ce cadre, le modèle des satisfactions peut être vu comme un outil pour analyser/traiter certaines situations. Mais il nécessite tout de même de manipuler trois dimensions : les position des agents, leur évolution dans le temps, les satisfactions.

Le principe de résolution proposé dans ce chapitre a été expérimenté et validé sur de véritable mini-robots mobiles. Ces expériences ont été menées sur le problème d'exploration d'un corridor étroit et fermé. Elles sont décrites au chapitre 12 et le protocole de communication employé (émissions de signaux par les agents) est défini au chapitre 11.



# Chapitre 10

## Systeme hétérogène et introduction de l'apprentissage

### 10.1 Introduction

Ce chapitre a pour double objectif de montrer que le modèle Satisfaction-Altruisme s'applique à des systèmes multi-agents hétérogènes, mais aussi que le modèle des satisfactions peut être utilisé efficacement dans le cadre de robots apprenants. Il s'agit en effet d'exploiter les communications locales de satisfactions interactives pour récompenser les actions des agents.

#### 10.1.1 Hétérogénéité des robots et des objets

**Le transport collectif et la découpe de déchets toxiques** Ce problème est une variante de l'énoncé des robots fourrageurs, qui introduit des agents aux fonctionnalités différentes et des objets de tailles quelconques. Les éléments à traiter ne sont plus composés d'un matériau homogène mais représentent des déchets toxiques (des plaques rectangulaires) de tailles très variables. Un premier type de robots peut détecter ces déchets et les déplacer jusqu'à la base : soit en les *poussant* soit en les *tirant*. Toutefois, leur force individuelle est limitée de telle sorte qu'ils ne peuvent déplacer isolément que des plaques de petites tailles. Par contre, ils peuvent agir simultanément à plusieurs sur la même plaque pour établir une force collective plus importante. Un deuxième type d'agents complète le système : ce sont des robots capables de détecter et *découper* les plaques, c'est-à-dire de les décomposer en plaques plus légères, donc plus facilement déplaçables.

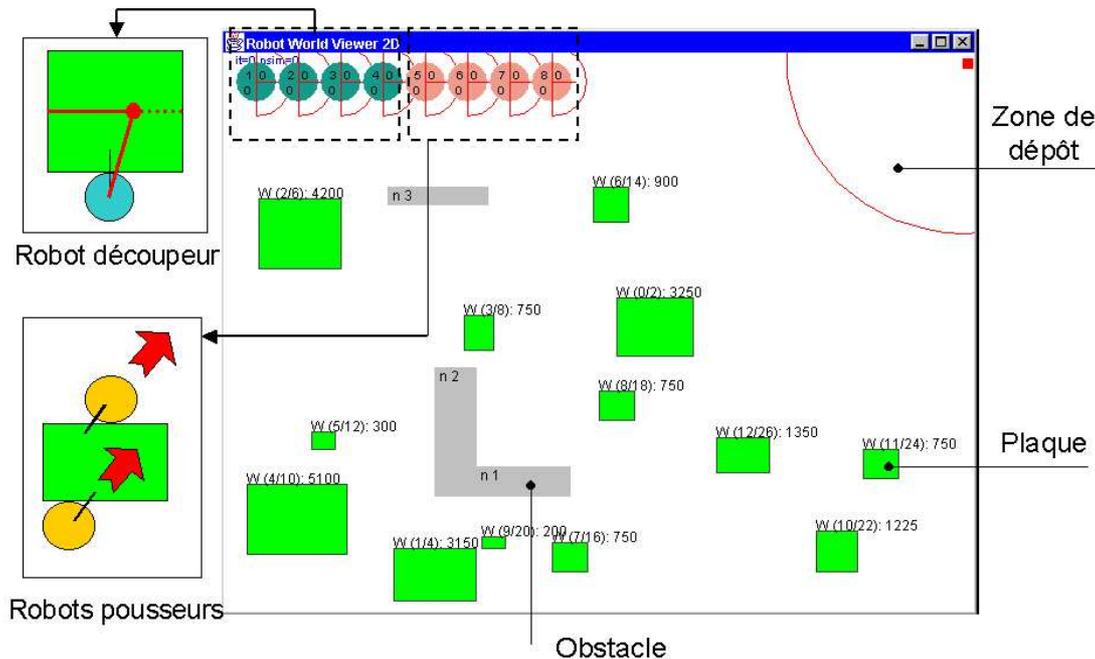


FIG. 10.1 – Copie d'écran de l'état initial d'une simulation et schématisation des robots découpeurs et des robots pousseurs.

**Définition et Représentation du problème** La figure 10.1 donne un aperçu général du problème abordé, il s'agit d'une copie d'écran du simulateur développé pour cette étude (cf. noyau du simulateur en section 8.3.1). Son développement a largement bénéficié du travail de DEA de J. Chapelle [Chapelle, 2001].

Précisons les divers types d'agents présents dans le problème :

- *Les robots pousseurs/tireurs* : Ils sont munis d'un bras qui leur permet de tracter ou de pousser une plaque, ou de la maintenir en place. Leur charge tractable est limitée. Ils sont également capables de se diriger vers la zone de dépôt. Une coopération indirecte est prévisible entre robots pousseurs/tireurs : les actions indépendantes mais simultanées de plusieurs robots peuvent se cumuler pour permettre le déplacement de plaques trop lourdes pour un seul agent (situation type box pushing).
- *Les robots découpeurs* : Ils sont munis d'une "scie" qui leur permet de manipuler et de découper une plaque en deux parties égales. Ils possèdent également un bras qui leur permet de maintenir la plaque pendant la découpe. Une plaque n'est découpée que si elle possède une taille suffisamment grande.
- *La base* : C'est un agent fixe qui émet un signal particulier permettant aux agents mobiles de la repérer.

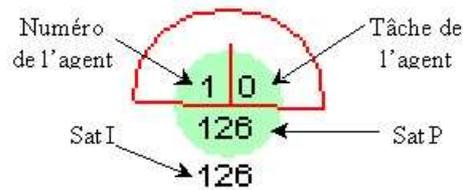


FIG. 10.2 – Informations affichées pour chaque robot simulé

Les règles de comportement des agents et d'évolution de l'environnement :

- *L'environnement* dans lequel évoluent les robots est rectangulaire et non cyclique, les bords sont considérés comme des murs infranchissables. L'environnement contient des obstacles fixes de forme rectangulaire.
- *La zone de dépôt* est une partie de l'environnement vers laquelle les robots doivent acheminer les plaques. Elle est représentée par un cercle autour de la base. Lorsqu'une plaque se trouve à l'intérieur de cette zone, elle est retirée (simulation d'un broyeur ou d'une fosse). Les robots perçoivent un signal émis en continu par la base.
- *Les plaques* : Chaque plaque peut être déplacée sous l'influence d'une force qui lui est appliquée. La force doit être supérieure au poids de la plaque pour en permettre le mouvement. Le poids est déduit directement de la surface de la plaque.
- *Déplacement des agents* : Les robots se déplacent à vitesse constante. Ils peuvent s'orienter dans n'importe quelle direction. Dans les simulations présentées, les robots sont de forme cylindrique (diamètre 34 cm). Lors d'une phase de recherche, ils adoptent un déplacement aléatoire.
- *Perception des agents* : Les robots perçoivent l'environnement sur 180 degrés devant eux et sur un rayon  $R_v$  de 30 cm, qui est représenté par un demi cercle dans les simulations, cf. fig. 10.2 (1 cm  $\simeq$  largeur d'un pixel dans les simulations).
- *Signaux émis par les agents* : Les signaux de satisfaction interactive sont émis sur une portée rayon  $R_c$  de 70 cm. Lors d'un appel de recrutement, ce rayon est augmenté à 100 cm.

Représentation graphique :

Afin de suivre l'évolution des robots, plusieurs paramètres sont affichés (cf. fig. 10.2) : le numéro identificateur de l'agent, celui de sa tâche en cours, le niveau de satisfaction personnelle (affiché à l'intérieur du robot) et la valeur de satisfaction interactive émise (affichée à l'extérieur du robot). Enfin, le type d'agent est représenté par sa couleur : gris foncé pour les robots découpeurs, gris clair pour les robots pousseurs.

## 10.2 Simulations et comportements coopératifs

### 10.2.1 Paramètres

Tableau d'évaluation de la satisfaction personnelle des fonctions des agents (cette valeur évolue dans l'intervalle  $[-127, 127]$ ) :

fonction	robot pousseur	robot découpeur
recherche aléatoire	si déplacement $v = 2.(0.5 + \cos(\alpha))$ sinon (blocage) $v = -3$	idem pousseur
déplacement plaque	si plaque en mouvement $v = 20$ sinon $v = -20$	Interdit
découpe plaque	Impossible	découpe sans aide $v = 10$ découpe avec aide $v = 20$ impossible $v = -40$

$\alpha$  est l'angle déterminé par la différence entre la direction réellement effectuée par l'agent et la direction de son but.

Le déplacement des plaques peut dans certains cas être impossible : poids trop important ou obstacles sur le parcours. Dans le cas d'une plaque trop lourde, l'agent devient insatisfait et suivant un certain seuil il émet un signal attractif de recrutement. Cet appel a lieu jusqu'à ce que la plaque puisse être déplacée ou si son insatisfaction le fait abandonner la tâche.

La découpe d'une plaque par un robot peut elle aussi être entravée : tout autre robot qui tente de découper ou pousser la plaque ralenti ou empêche sa découpe.

Le poids de déclenchement des fonctions découper et pousser est égal à 0.5, celui de l'exploration aléatoire vaut 0.1.

Signaux émis par les agents :

situation	robot pousseur	robot découpeur
plaque trop lourde à déplacer	appel d'aide $I(t) = 127$ (pour une durée limitée)	
gêne pendant déplacement plaque	répulsion $I(t) = P(t)$	
gêne pendant découpe d'une plaque		répulsion $I(t) = P(t)$

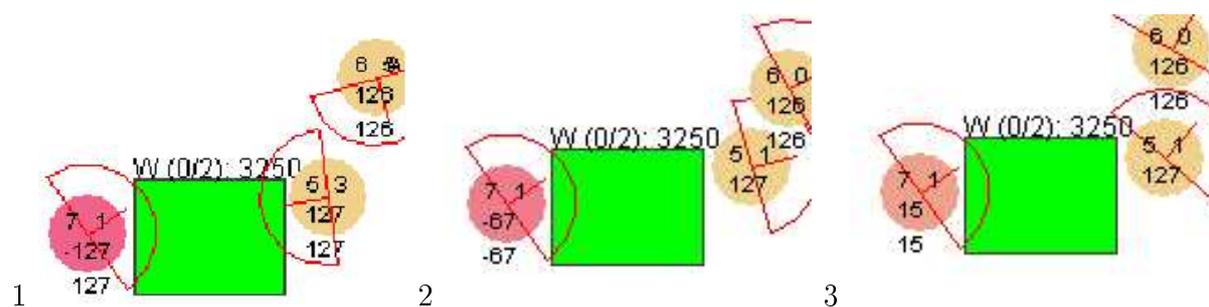


FIG. 10.3 – (1) l'agent 7 ne peut déplacer la plaque seul, il émet un signal de recrutement, l'agent 5 vient de découvrir la plaque, il émet aussi un signal attractif, (2) les agents 7 et 5 parviennent à déplacer la plaque, l'agent 6 tente de les éviter, (3) le déplacement progresse vers la zone de dépos, la satisfaction de l'agent 7 redevient positive.

### 10.2.2 Simulations

**Observations générales** Nous avons effectué des simulations sur des environnements relativement variables (faible à forte densité des agents ou des obstacles, ex. fig. 10.1, 10.7). Les comportements obtenus sont parfaitement cohérents et assurent la réalisation de la tâche globale.

Comme pour les robots fourrageurs (chapitre 8), les simulations débutent par une exploration de l'environnement relativement efficace, car les agents ont tendance à se repousser. Puis, lorsque les agents découvrent des plaques ils agissent immédiatement, soit ils les découpent, soit ils les transportent vers la zone de dépôt, et ce jusqu'à ce qu'il ne reste plus de plaque.

#### Coopérations

La simulation confirme les comportements coopératifs attendus :

- directs : appel d'aide/recrutement lorsqu'un pousseur est incapable d'agir seul,
- indirects : cumul des forces d'agents pousseurs/tireurs indépendants pour déplacer une plaque (cf. exemple fig. 10.3).

#### Conflits

Lorsque un agent découpeur découvre une plaque libre de taille suffisante pour la traiter, il la découpe en deux nouvelles plaques. Il réitère l'opération jusqu'à la décomposer en plus petites plaques.

La figure 10.4 présente un tel processus mais en présence de deux robots découpeurs. Le robot 2 est insatisfait car le robot 1 découpe la plaque, ce dernier est aussi gêné par

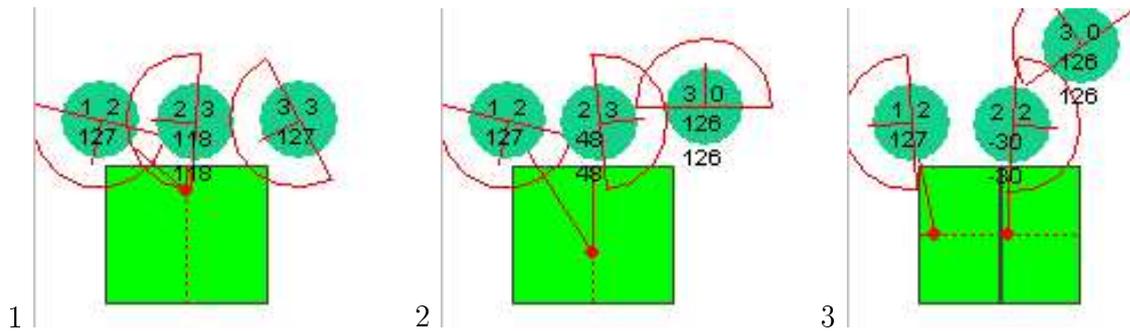


FIG. 10.4 – (1) début de découpe pour l'agent 1, (2) le découpeur 2 est insatisfait de ne pouvoir agir, (3) l'agent 1 a fini, les deux agents traitent chacun une des deux nouvelles plaques.

les tentatives du second. Ce conflit entre les deux agents s'arrête dès que la plaque est partagée ou que les insatisfactions deviennent très négatives.

D'autres conflits ont été observés lorsque des agents de types différents tentent simultanément des actions incompatibles sur la même plaque (découper / pousser). Les raisons essentielles à ces problèmes sont la portée parfois trop faible des signaux ou bien encore l'émission de signaux d'appels par des agents pousseurs qui mènent au recrutement d'agents découpeurs. Ce dernier problème pourrait être traité de façon ad hoc en empêchant les agents découpeurs de réagir aux signaux attractifs.

### Vers une évolution du modèle

La définition de ce système hétérogène dans le modèle satisfaction-altruisme n'a pas posé de problèmes particuliers. Chaque agent agit en fonction de ses perceptions locales et des signaux qu'il perçoit, ces derniers garantissant la cohérence des trajectoires et des interactions des agents (jusqu'à une certaine mesure, comme nous venons de le voir). La seule exigence du modèle, comme pour tout système hétérogène communicant, est le partage du système de communication commun.

Nous avons observé aux cours des simulations des comportements coopératifs très efficaces, mais aussi certaines interactions insatisfaisantes. L'existence de ces problèmes nous a amené à envisager une exploitation différente des états de satisfaction des agents. Ainsi, nous avons entrepris d'utiliser la satisfaction interactive pour intégrer une méthode d'apprentissage au modèle. Le but étant d'étendre son adaptativité pour traiter les conflits présentés précédemment.

## 10.3 Introduction de l'apprentissage

### 10.3.1 Éviter les conflits, apprendre à coopérer

La redondance des conflits présentés auparavant provient de l'incapacité du modèle à retenir des comportements se révélant efficaces, mais aussi parce que la réaction altruiste n'est pas toujours le comportement optimal à certaines situations (fuir un agent insatisfait pourrait être remplacé par une action d'aide plus profitable).

L'idée générale consiste donc à introduire une alternative à la réaction altruiste. Pour cela, l'agent doit pouvoir essayer d'autres comportements, évaluer leurs effets et retenir les plus efficaces.

Notre démarche est plus particulièrement motivée par l'utilisation des satisfactions pour établir la *récompense* des expériences. Cependant, le modèle des signaux utilisé jusqu'à présent se prête mal à une technique de ce type. En effet, les agents évaluent la progression de leur tâche pour déterminer leur niveau de satisfaction personnelle. Celle-ci, couplée à l'observation des actions des agents voisins, peut mener au déclenchement de l'émission de signaux d'*influences*. Ces influences provoquent chez les agents qui les perçoivent des réactions altruistes *pré-définies*. Cette technique de réaction systématique est donc à l'opposée d'un comportement apprenant, qui consiste à prendre des initiatives pour tester de nouveaux comportements et retenir les plus pertinents.

L'introduction d'une alternative à la réaction altruiste est donc nécessaire à l'intégration d'une capacité d'apprentissage au modèle. De plus, il faut permettre aux agents d'expérimenter/sélectionner des comportements différents pour une même situation et qu'ils obtiennent en retour une récompense.

Nous avons vu que les signaux échangés par les agents étaient des formes de requêtes passives (ou influences). L'emploi de ces signaux comme valeur de récompense n'est donc pas possible, puisqu'ils ne représentent pas l'évaluation d'actions, mais leur conséquence. Par contre, la satisfaction personnelle représente cette notion. Ainsi, dans le cadre de l'apprentissage, les signaux émis sont définis par cette satisfaction (cf. 10.3.3).

Toutefois, dans cette démarche d'intégration de l'apprentissage, nous n'avons pas complètement fait disparaître la notion d'influence, car nous avons montré l'intérêt d'utiliser des attractions (chapitre 8) et des répulsions (chapitre 9). Aussi, nous conservons cette notion mais de façon limitée : les influences sont codées seulement par les valeurs extrêmes de l'intervalle d'émission, ceci est représenté en figure 10.5. Les valeurs émises dans l'inter-

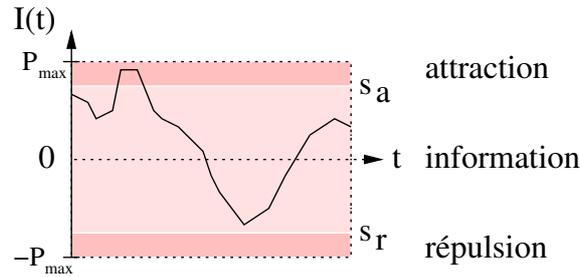


FIG. 10.5 – Sémantique des signaux émis dans le modèle avec apprentissage

valle  $[s_r, s_a]$  sont égales au niveau de satisfaction interactive de l'agent, qui ne sont plus des influences mais des états de l'agent.

Dans le but d'évaluer l'utilisation des signaux comme valeurs de récompenses, nous avons intégré à l'architecture une technique d'apprentissage par renforcement classique. Elle consiste à modifier, en fin d'expérience, le poids de déclenchement des fonctions exécutées. Comme les expériences peuvent être "longues", nous ne tenons pas compte, dans le renforcement, de la situation de l'agent en fin d'exécution de la fonction (approche type [Mataric, 1998]).

Depuis quelques années, l'apprentissage dans les systèmes distribués est devenu un thème de recherche à part entière [Sen and Weiss, 2000], cf. classification détaillée des différentes approches dans [Drogoul and Zucker, 1998]. Dans ce cadre, l'emploi de communications simples (bas niveau) est une approche pour améliorer l'apprentissage d'agents situés autonomes. Notons en particulier les travaux de [Parker, 1995], [Shibata et al., 1996], [Mataric, 1998], [Mataric, 2001]; les enjeux et problèmes à traiter sont précisés dans [Sen and Weiss, 2000] p. 285.

Notre objectif est de montrer que les signaux de satisfaction interactive peuvent permettre d'intégrer facilement des capacités d'apprentissage à un modèle d'agents autonomes simples. Notre employons une démarche expérimentale (par simulation) pour évaluer l'apparition de comportements coopératifs. Les signaux de satisfaction sont utilisés à deux niveaux : (i) pour définir un espace d'états des agents qui soit de faible coût mémoire, (ii) pour calculer les valeurs de récompense des actions exécutées.

### 10.3.2 Intégration de l'apprentissage dans le modèle

La figure 10.6 résume le principe d'intégration de l'apprentissage dans le modèle. D'une part, un agent ne réagit plus systématiquement par altruisme ou par la fonction de poids

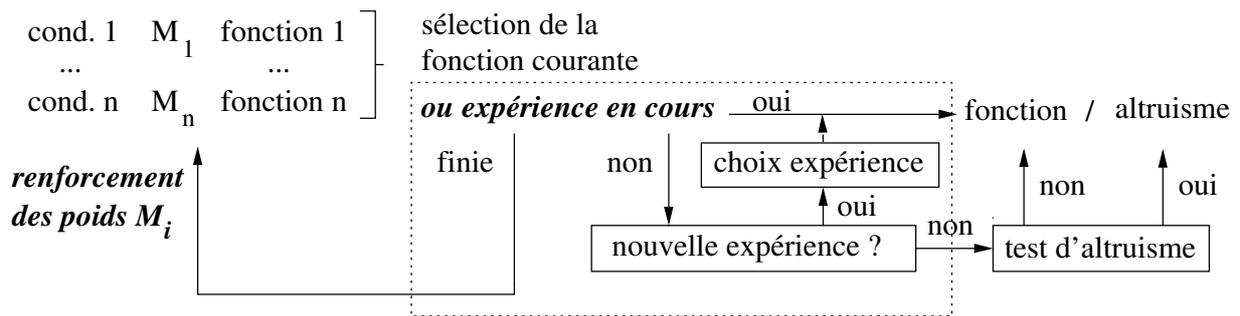


FIG. 10.6 – Intégration d'un module d'apprentissage par renforcement dans le modèle satisfaction-altruisme.

maximum, mais peut déclencher une expérience. D'autre part, les résultats de ces expériences modifient le poids de déclenchement des fonctions (partie renforcement). La technique proposée est *on-policy*, c'est-à-dire qu'elle permet d'alterner des phases d'apprentissage (des expériences) et des phases d'exploitation.

L'intégration de l'apprentissage a nécessité l'ajout des éléments suivants :

- Ajout au test de sélection d'une éventuelle expérience, c'est-à-dire de l'exécution d'une fonction ou de l'altruisme pendant une durée déterminée.
- Exécution d'une opération de renforcement en fin d'expérience, dont la valeur est une fonction de la moyenne des satisfactions perçues pendant l'exécution.

#### Conditions de déclenchement d'une expérience :

Pour décider de l'exécution d'une expérience, un agent doit se trouver dans une situation respectant les conditions suivantes :

- ne pas déjà être en phase d'exécution d'une expérience,
- se trouver en phase de changement de fonction, ou bien percevoir des signaux négatifs.

Si ces deux conditions sont vérifiées alors l'agent décide, suivant une probabilité fixée, soit d'un comportement d'*expérimentation* soit d'un comportement d'*exploitation* (modèle classique, exécution de la tâche de poids le plus fort ou altruisme).

Si l'expérimentation est décidée, l'agent sélectionne une fonction parmi celles dont il dispose (dont l'altruisme). La probabilité de sélection d'une fonction est inversement proportionnel à son nombre d'essais. Cette pondération a pour but d'assurer l'exploration de tous les comportements possibles (cf. détails d'implémentations dans [Chapelle, 2001]).

#### Calcul de la récompense et du renforcement :

Un agent évalue au cours d'une expérience *la valeur moyenne des satisfactions* qu'il perçoit. S'il perçoit plusieurs signaux, il sélectionne le signal le plus négatif, sinon le plus

positif. Dans le cadre de l'apprentissage, cette priorité exprime que l'agent ne renforce que les fonctions qui satisfont *tous* ses voisins, et pas seulement une partie d'entre eux.

Cette moyenne, noté  $\overline{SatI}$ , est directement utilisée dans la formule de renforcement. Toutefois, une valeur moyenne peut ne pas être significative de l'amélioration d'une situation. Imaginons un agent réagissant à une interaction par l'émission de valeurs évoluant des plus négatives aux plus positives, alors la moyenne perçue pourrait être nulle, bien que l'interaction ait une conséquence très positive. Pour tenir compte de cette notion de variation relative, le calcul de la récompense intègre la différence entre la valeur moyenne des satisfactions perçues et la première valeur perçue ( $SatI_0$ ). Cette différence est définie par  $\Delta = \frac{1}{2.P_{max}}.(\overline{SatI} - SatI_0)$ , ainsi  $\Delta \in [-1, 1]$ .

La récompense  $r$  d'une expérience tient compte de cette variation si elle est importante (grandeur de  $|\Delta|$ ), sinon elle prend en compte essentiellement la valeur moyenne. La récompense se calcule ainsi :

$$r' = |\Delta| . \Delta + (1 - |\Delta|) . \frac{\overline{SatI}}{P_{max}}$$

donc  $r' \in [-1, 1]$ , il faut normaliser cette récompense à l'intervalle de valeur des poids de motivation, c'est-à-dire sur  $[0, 1]$  :

$$r = \frac{1}{2}(r' + 1)$$

$r \in [0, 1]$ , alors le poids de la fonction expérimentée est renforcée classiquement :

$$M_i = \beta.M_i + (1 - \beta).r$$

( $\beta$  vaut 0.9 dans les simulations).

### Représentation des situations

Nous utilisons à nouveau les signaux de satisfactions pour décrire l'état (la situation) d'interaction des agents. En effet, jusqu'à présent nous n'avons déterminé les conditions ou situations de déclenchement des comportements que par des règles pré-définies (ou perceptions).

Généralement, les états (ou situations d'actions) des agents sont définis par combinaison de différentes perceptions pertinentes vis à vis du problème traité (reconnaissance de scènes, d'objets, d'agents voisins, états des capteurs extéroceptifs). Cette démarche classique, dans l'emploi d'une méthode d'apprentissage par renforcement, s'expose au problème de l'explosion combinatoire du nombre de situations potentiellement rencontrables. Nous avons exploité la sémantique des signaux de satisfaction pour tenter de limiter ce problème.

Notre approche consiste à déterminer de façon minimale les interactions entre agents (actions conflictuelles ou coopérantes) en ne considérant que les signes des valeurs échangées. Cependant, ces signaux ne contiennent pas d'information sur le *type* des agents qui les émettent. Pourtant cette information est cruciale dans la détermination d'une situation mettant en jeu un groupe hétérogène d'agents.

Nous avons alors introduit dans les signaux échangés un bit d'information sur la nature de l'agent émetteur (0 : découpeur, 1 : pousseur). Ainsi, nous avons obtenu un ensemble de situations-actions contenues dans une matrice de taille raisonnable (présentée ci-après).

### 10.3.3 Simulations et analyse des résultats

#### Matrice des comportements

Pour mettre en valeur le système d'apprentissage intégré au modèle, nous reprenons le problème des robots découpeurs et pousseurs de plaques. Rappelons l'ensemble des fonctions dont ils disposent :

- 0 : recherche aléatoire d'une plaque,
- 1 : pousser ou tirer une plaque,
- 2 : découper une plaque (impossible pour les agents pousseurs),
- 3 : approcher d'une plaque détectée,
- 4 : maintien d'une plaque (l'agent saisit seulement la plaque),
- 5 : déplacement altruiste : fuite ou approche.

La fonction - maintien d'une plaque - a été introduite pour permettre à un agent d'aider une opération de découpe. En effet, le maintien d'une plaque accélère la vitesse de découpe, donc la satisfaction de celui qui réalise cette opération. Ceci dans le but de vérifier l'émergence de comportements coopératifs autour de la découpe des plaques.

Le calcul des niveaux de satisfaction personnelle des agents, défini en section 10.2.1, est dépendant des interactions entre agents (sauf pour les déplacements). Ainsi, nous pouvons poser pour ce problème  $I(t) = P(t)$ . Les valeurs émises sont comprises dans l'intervalle  $[-126, +126]$  (+127 est réservé à l'appel d'aide d'un robot pousseur/tireur).

La *situation courante* d'un agent est définie à partir de deux types de perceptions : les valeurs des signaux reçus et l'éventuelle perception d'une plaque (absence, perception, plaque saisie). Pour réduire l'espace des valeurs des signaux, nous ne considérons que leur signe et le type d'agent qui les émettent.

Les situations sont codées par trois variables ternaires (I1, I2, S3) :

code	I1 : signal d'un découpeur	I2 : signal d'un pousseur	S3 : état plaque
<b>0</b>	aucun	aucun	aucune
<b>1</b>	de signe +	de signe +	perçue
<b>2</b>	de signe -	de signe -	plaque saisie

Ces trois variables nous permettent de représenter en seulement 27 cas différents, les situations “interactives” de chaque agent : la combinaison de la sémantique des signaux émis par chaque type de voisins et la relation de l'agent avec une éventuelle plaque.

Cette représentation nous a conduit à définir une matrice des comportements, c'est-à-dire des poids déclencheurs de chaque couple (situation, fonction) pour chaque type d'agent (technique employée par M. Mataric par exemple pour des robots autonomes non communicants [Mataric, 1994a]). Les deux matrices résultantes pour chaque type d'agent sont données avec leurs valeurs initiales :

		Pousseur/tireur																											
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
T		000	001	002	010	011	012	020	021	022	100	101	102	110	111	112	120	121	122	200	201	202	210	211	212	220	221	222	
0	Chercher	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
1	Pousser			0,5						0,5						0,5						0,5							0,5
2	Decouper																												
3	Approcher		0,5			0,5				0,5			0,5			0,5			0,5			0,5			0,5			0,5	
4	Maintenir			0,5			0,5			0,5			0,5			0,5			0,5			0,5			0,5			0,5	
5	Altruisme				0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6

		Découpeur																											
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
T		000	001	002	010	011	012	020	021	022	100	101	102	110	111	112	120	121	122	200	201	202	210	211	212	220	221	222	
0	Chercher	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
1	Pousser																												
2	Decouper			0,5			0,5			0,5			0,5			0,5			0,5			0,5			0,5			0,5	
3	Approcher		0,5			0,5				0,5			0,5			0,5			0,5			0,5			0,5			0,5	
4	Maintenir			0,5			0,5			0,5			0,5			0,5			0,5			0,5			0,5			0,5	
5	Altruisme				0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6

Les tables sont initialisées suivant les comportements définis par le modèle satisfaction-altruisme sans apprentissage (cf. section 10.2.1). La fonction altruisme de chaque agent reçoit un poids de 0.6 lorsqu'un quelconque signal est perçu (le test d'altruisme décidant ensuite de son exécution réelle). Toutes les situations d'applications évidentes des fonctions/actions sont volontairement initialisées à la même valeur 0.5, dans le but de laisser le processus d'apprentissage créer des variations. Ainsi, pour toute situation de saisie d'une plaque par un pousseur l'action pousser/tirer reçoit la valeur 0.5, idem pour la fonction découpe d'un découpeur, ainsi que la fonction approcher pour tout agent percevant une plaque qu'il n'a pas encore saisie. Les deux types d'agents reçoivent aussi un poids initial de 0.5 pour la fonction maintenir dans les situations où ils saisissent une plaque.

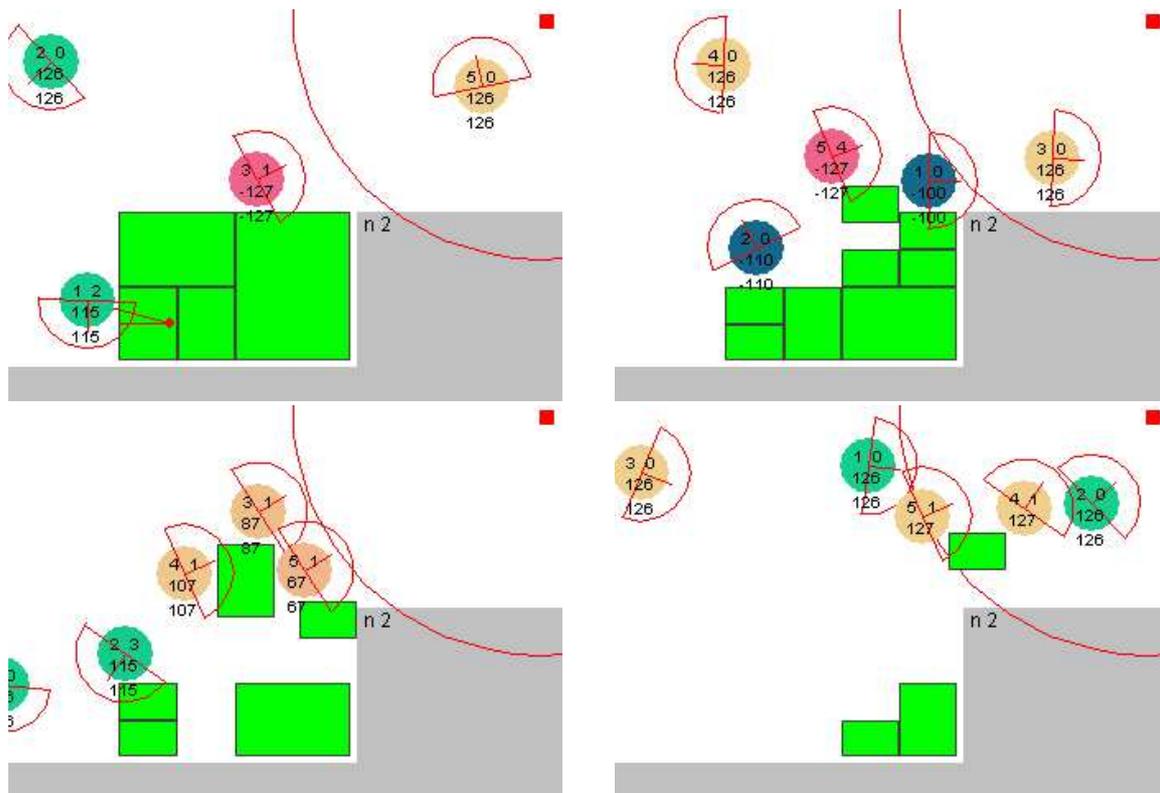


FIG. 10.7 – Évolution d'une simulation.

Enfin, toute situation reçoit par défaut un faible poids de 0.1 pour la recherche d'une plaque dans l'environnement. Les cases vides ont pour valeur 0, les cases grises expriment une fonction n'existant pas chez l'agent.

### Résultats et validation du principe

Ces agents apprenants ont été programmé et testé sur divers types d'environnements en collaboration avec J. Chapelle durant son DEA [Chapelle, 2001]. Nous présentons ici les résultats obtenus sur un environnement conçu pour maximiser les chances de provoquer des interactions entre les agents. Ce système, nommé "bloc12600" est représenté en figure 10.7.

L'intérêt d'utiliser un tel environnement est de provoquer des gênes entre plusieurs robots découpeurs, puisqu'il n'y a qu'un seul objet. Mais aussi de créer des interactions entre agents pousseurs/tireurs et découpeurs puisque la plaque n'est accessible que par seulement deux de ses côtés.

Les résultats obtenus ne peuvent être analysés qu'en examinant les matrices après plusieurs résolutions successives du problème (ici 72). Nous avons tout de même observé à l'écran un certain nombre de comportements, en dehors des phases d'apprentissages, qui n'étaient pas prévus par les matrices initiales. Nous présentons ci-dessous les résultats les plus significatifs sur cet environnement.

Notons tout d'abord que ces résultats ont été obtenus en très peu d'essais, les comportements les plus pertinents ont été appris *entre trente et quarante essais* de la même fonction.

### Robot pousseur (extrait des matrices du robot pousseur n°4)

découpeur			pousseur			plaque		
P+ P-			P+ P-			perc. saisie		
0	1	2	0	1	2	0	1	2

matrice des poids déclencheurs

Agent n4	000	001	002	010	011	012	020	021	022	100	101	102	110	111	112	120	121	122	200
Chercher	0,10	0,10	0,10	0,66	0,19	0,10	0,19	0,11	0,10	0,81	0,10	0,10	0,66	0,10	0,10	0,30	0,10	0,10	0,33
Pousser			0,50			0,72			0,16			0,67			0,88			0,36	
Découper																			
Approcher		0,50			0,66		0,00	0,45			0,61			0,55			0,50		
Maintenir			0,50			0,66			0,14			0,53			0,57			0,42	
Altruisme				0,60	0,55	0,60	0,20	0,21	0,12	0,60	0,60	0,60	0,60	0,60	0,60	0,38	0,46	0,45	0,42

Agent n4	000	001	002	010	011	012	020	021	022	100	101	102	110	111	112	120	121	122	200
Chercher	1	1	1	18	2	1	<b>127</b>	3	5	<b>23</b>	1	1	11	1	1	<b>18</b>	1	1	<b>53</b>
Pousser			1			12			<b>53</b>			9			7				5
Découper																			
Approcher		1			5		1	2			7			2			1		
Maintenir			1			5			<b>52</b>			7			4				5
Altruisme				1	2	1	<b>425</b>	<b>19</b>	<b>54</b>	1	1	1	1	1	1	<b>80</b>	4	7	<b>168</b>

matrice des nombres d'essais

- Examinons la situation 012, l'agent pousseur est accroché à une plaque et perçoit des signaux positifs d'un autre robot pousseur. Cette situation à provoquer le renforcement de la fonction pousser (passant de 0.5 à 0.72), qui correspond bien à un comportement d'entre-aide. Par contre, dans la situation 022, l'agent n'a pas appris à aider un pousseur en difficulté (nous revenons sur ce problème plus loin).

- Nous pouvons constater que l'agent a souvent déclenché l'état d'altruisme en tant qu'expérience. Le résultat est intéressant puisque pour les situations où un robot pousseur voisin a émis une insatisfaction (021, 022, 120, 121, 122) l'agent a diminué son poids de réaction de fuite. En effet, cette réaction ne permet pas de répondre au besoin d'aide évident des voisins.

## Robot découpeur

matrice des poids déclencheurs

Agent n1	000	001	002	010	011	012	020	021	022	100	101	102	110	111	112	120	121	122	200	201	202	210	211	212	220	221	222
Chercher	0,10	0,10	0,10	0,69	0,32	0,10	0,10	0,14	0,11	0,70	0,16	0,10	0,62	0,10	0,10	0,20	0,10	0,10	0,29	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10
Pousser						0,17			0,11			0,40			0,13			0,02			0,11						
Découper			0,50			0,66			0,09			0,66			0,54			0,38			0,44			0,50			0,50
Approcher		0,50			0,83		0,18	0,10			0,78			0,65		0,01	0,38		0,01	0,25			0,50			0,50	
Maintenir			0,50			0,53			0,15			0,62			0,59			0,42			0,37			0,50			0,46
Altruisme				0,55	0,60	0,57	0,19	0,10	0,10	0,60	0,60	0,60	0,60	0,60	0,60	0,19	0,44	0,41	0,36	0,18	0,43	0,60	0,60	0,60	0,18	0,50	0,55

Agent n1	000	001	002	010	011	012	020	021	022	100	101	102	110	111	112	120	121	122	200	201	202	210	211	212	220	221	222
Chercher	1	1	1	22	4	1	312	15	3	24	2	1	13	1	1	29	1	1	50	4	1	1	1	1	13	1	1
Pousser						3			35			8			2			3			5						
Découper			1			7			37			5			2			4			5			1			1
Approcher		1			18		7	52			15			6		1	4		1	25			1			1	
Maintenir			1			7			40			4			5			3			9			1			2
Altruisme				2	1	2	606	103	30	1	1	1	1	1	1	69	7	5	171	23	7	1	1	1	22	3	2

matrice des nombres d'essais

Comme pour le robot pousseur, ce robot découpeur a effectué des renforcements négatifs pour des actions incompatibles avec celles de ses voisins (022, 122, 202 découper et maintenir).

Les renforcements positifs sont plus rares. Toutefois, examinons la situation 122 : l'agent découpeur a saisi une plaque, il perçoit un signal positif d'un autre découpeur et un signal négatif d'un voisin pousseur/tireur. Les différents renforcements ont amené l'agent à mettre l'action maintenir prioritaire sur les autres. C'est un comportement coopératif, puisque en réalisant cette action il permet à son voisin d'effectuer sa tâche de découpe avec un niveau de satisfaction plus élevé.

Enfin notons qu'en situation 022 (perception d'un voisin pousseur insatisfait) l'agent découpeur a fait apparaître une nouvelle fonctionnalité dans ses possibilités d'actions. En effet, la tâche pousser (de poids initial 0) est devenu, après 35 essais, une des fonctions de plus fort poids. Cependant, il faut relativiser la qualité de ce résultat par la faiblesse de la valeur obtenue (seulement 0.11). Ceci s'explique par le fait que certaines situations théoriquement coopératives n'ont pas produit d'émissions de signaux positifs : les agents peuvent ne pas être assez nombreux pour déplacer la plaque ou bien les agents peuvent se gêner spatialement. Ce type de situations non récompensées ralentit le processus d'apprentissage mais ne l'empêche pas.

## Limites du modèle proposé

Certains nouveaux comportements apparaissent de faible utilité, voire contradictoire. Par exemple le robot découpeur en situation 102 a renforcé sa fonction découper alors qu'il perçoit un autre découpeur satisfait. L'explication d'un tel renforcement nous est donnée par l'observation des simulations : les agents peuvent percevoir des signaux

provenant de voisins qui *ne travaillent pas sur la même plaque* que la leur (c'est très vrai sur l'environnement considéré). Ainsi un agent peut prendre en compte des signaux ne concernant pas ses actions (c'est un problème que nous avons évoqué dès le chapitre 5 et qui nous avait conduit à intégrer des signaux d'influences plutôt que d'états).

Les renforcements inefficaces ont deux sources :

- Le codage actuel des signaux ne permet pas à un agent de savoir si un signal exprime une (in)satisfaction vis à vis des actions qu'il mène. Il faudrait que les agents puissent établir une relation entre signaux, actions et objets manipulés. Pour cela, nous travaillons actuellement à définir plus finement les signaux émis, en donnant la possibilité aux agents de diriger leurs émissions (dans des secteurs).
- Le calcul de la récompense est basé sur une mesure des effets immédiats des actions exécutées. Cette démarche n'est pas valide pour les tâches nécessitant les efforts cumulés de plusieurs agents pour devenir satisfaisantes (par exemple le déplacement des plaques nécessitant plusieurs agents).

## 10.4 Bilan

- Nous avons montré en début de chapitre que le modèle satisfaction-altruisme était facilement applicable à un système hétérogène. Deux types d'agents aux fonctionnalités différentes ont ainsi été défini indépendamment. Des tests effectués par simulation, mettant en jeu plusieurs robots de ces deux types, montrent un comportement global cohérent, menant à la résolution du problème.

Remarquons que rien n'empêche d'imaginer la présence d'autres robots n'implémentant pas notre modèle. Dans ce cas, le module de navigation des agents "altruistes" assurerait un comportement d'évitement vis-à-vis de ces obstacles mouvants. De plus, la présence des autres agents ne remettrait pas en cause les influences échangées entre les agents exécutant le modèle satisfaction-altruisme.

- Nous avons vu que les systèmes hétérogènes provoquaient certains conflits et que la réaction altruiste (au sens du modèle) n'était pas toujours pertinente. Cette constatation nous a mené à intégrer un module d'apprentissage par renforcement au coeur de l'architecture. Les différentes simulations effectuées montrent l'apparition de nouvelles fonctionnalités coopérantes chez les agents, de l'amplification ou de la diminution à profit de certaines.

- L'utilisation du modèle des satisfactions s'est révélée doublement pertinente : d'abord par une modélisation compacte des interactions entre agents, en terme de perception des signaux et des objets. Mais aussi par l'utilisation de ces valeurs perçues pour calculer la récompense du processus de renforcement.

Les premiers résultats obtenus en simulations sont encourageants, mais montrent aussi des renforcements inutiles ou contradictoires. Nous travaillons actuellement à les réduire par une évolution des capacités de perceptions et d'émissions des agents, ainsi que par l'emploi d'une technique de Q-Learning.

- L'étude de cette intégration d'un module d'apprentissage, utilisant les notions de satisfactions, va être complétée et approfondie. En particulier les aspects temporels et de dépendances entre fonctions seront pris en comptes. Nous espérons alors faire évoluer cette étude vers une architecture potentiellement implémentable sur des mini-robots mobiles.



## Cinquième partie

# Communications et expérimentations en robotique mobile distribuée



# Chapitre 11

## Un protocole de communication pour SMA situé

### 11.1 Le problème

Ce chapitre présente le développement d'un protocole de communication dédié aux systèmes multi-robots autonomes mobiles. Notre objectif est d'implémenter un processus d'émission-réception par diffusion locale qui soit le plus performant et le plus robuste possible.

Le problème peut s'énoncer ainsi :

---

Spécificités d'un réseau de communication constitué de robots mobiles autonomes :

- il est formé de stations indépendantes, donc asynchrones (car *autonomie* des robots),
- il est dynamique car les robots sont *mobiles*,
- il peut être fortement chargé (communications en continu et simultanées entre plusieurs agents voisins),
- il n'est pas sûr (pertes de messages possibles),
- il est ouvert (intégrations et pertes de robots possibles dans le système),
- il est composé d'un nombre important d'agents (jusqu'à plusieurs dizaines),
- les messages sont émis sous formes de signaux omnidirectionnels de portée limitée.

Question : comment définir un protocole de communication décentralisé répondant à l'ensemble de ces contraintes ?

---

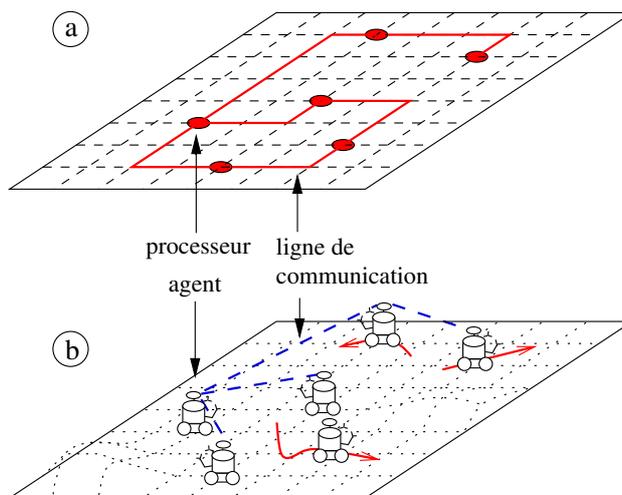


FIG. 11.1 – (a) un système distribué de processeurs (b) un système distribué de robots mobiles autonomes

### Réseaux de machines et SMA situés

La figure 11.1 schématise le type de “réseau dynamique” que nous considérons. Le tableau ci-dessous donne les différences essentielles entre les deux types de systèmes.

	Machine	relation	Robots mobiles autonomes
type de station	processeur	$\subset$	processeur + capteurs + effecteurs
stabilité géographique	totale	$\neq$	déplacements + comm. locales = forte dynamique
médium de communication	câbles (sûr)	$\neq$	signaux/ondes (peu sûr)
distance de communication	quelconque	$\supset$	locale

Ce qui différencie un réseau de robots mobiles par rapport à un réseau de machines fixes est en premier lieu sa mobilité physique (ou dynamique géographique). Le graphe du réseau des robots est constamment en variation et des arrivées-départs-pannes d’agents peuvent se produire à tout moment. La deuxième différence d’importance provient de la limitation des communications à des diffusions locales. Mais ceci nous permettra de considérer, dans certains cas, de petits groupes d’agents en communications plutôt que l’ensemble des robots du système (impliquant des problèmes de réseaux, ex. [Cohen et al., 1998])

#### 11.1.1 Choix technologiques

Pour mettre en place un système de communication dans un réseau dynamique et ouvert, il faut que les agents aient un accès simple et unique au médium de communication.

En effet, chaque nouvel agent entrant dans le système doit pouvoir immédiatement percevoir les messages de ses voisins et leur transmettre des informations, sans que ceux-ci ne modifient leur processus de communication.

Cette contrainte, présente dans les réseaux locaux de machines (LAN Local Area Network), implique le partage d'un unique médium de communication. Pour des robots autonomes, cela se traduit par l'utilisation d'une seule et même fréquence d'émission-réception par tous les agents. Ainsi, tout nouvel agent entrant dans le système doit seulement détenir un émetteur-récepteur sur cette fréquence pour pouvoir "communiquer" avec le reste du système.

**La couche liaison de données** Rappelons que notre modèle fonctionne sur la diffusion locale de simples valeurs. Cette forme de diffusion par signaux implique un service *sans connexion*<sup>1</sup> et *sans accusé de réception*. Si une valeur est perdue, il n'est prévu aucun moyen de remédier à cette perte. Ce type de service convient lorsque le taux d'erreur est faible ([Tanenbaum, 1998] p. 186). Il est très utilisé dans les réseaux locaux, ainsi que dans le cas de trafics en temps réel (pour lesquels il vaut mieux avoir des données légèrement altérées que des données en retard).

### 11.1.2 Caractéristiques du protocole recherché

De l'ensemble des contraintes données dans l'énoncé du problème général, nous avons dégagé deux principes que le protocole doit respecter pour être performant (tout au moins pour répondre aux contraintes).

#### **Principe d'équité (de communication)**

Protocole qui garantit à chaque agent l'égalité d'accès et d'utilisation du moyen de communication.

#### **Principe de robustesse (du protocole de communication)**

Protocole qui est capable de conserver ses caractéristiques malgré la perte ou l'intégration d'agents dans le système.

Corollaire :

**Robustesse d'un protocole de communication équitable** : protocole qui conserve le principe d'équité lors de l'intégration ou de la perte d'agents.

L'application du premier principe (*équité*) permet d'assurer à chaque agent un accès régulier au réseau, lui évitant ainsi des situations de famines. Ce choix d'équité résulte de l'absence

---

<sup>1</sup>Aucune connexion n'est établie au préalable, ni libérée après l'envoi des données.

de priorité entre les agents et de la recherche d'un protocole ouvert et auto-adaptatif aux besoins des agents.

Des protocoles existants proposent des implémentations de ce principe, soit au niveau *couche haute*, soit au niveau *couche physique*. Les solutions au niveau couche haute utilisent des techniques de réservations, de jetons, etc [Tanenbaum, 1998]. Nous n'avons pas retenu ce type d'approche car elle n'est pas du tout adaptée aux réseaux dynamiques, ni à la diffusion locale d'informations.

La deuxième solution (traitement au niveau couche physique) a vu un certain nombre de réalisations dans les réseaux de machines locales (techniques de multiplexage), nous les présentons ci-dessous. Toutefois, ces solutions comportent des restrictions les rendant mal adaptées à des agents autonomes. Ainsi, nous avons entrepris de nous en inspirer pour développer un protocole dédié aux SMA situés.

Le deuxième principe (*robustesse*) tient compte des variations possibles du nombre d'agents formant le réseau de communication. L'objectif étant d'établir un protocole conservant ses caractéristiques, en particulier le principe d'équité, lorsque ce type de variation survient.

## 11.2 Les protocoles existants

### 11.2.1 L'approche supervisée

Le moyen le plus simple de faire communiquer des robots mobiles est de traiter le problème de manière centralisée. L'approche consiste à connecter chaque agent à une machine centralisant tous les échanges de messages. Évidemment cette approche ne respecte plus les principes d'autonomie et de robustesse que nous nous sommes fixés. Citons trois exemples de ce type d'implémentations :

- Le projet Martha du LAAS [Alami et al., 1998],
- Les robots “autonomes” de L. Parker utilisant pourtant un système de communication centralisé [Parker, 1999] (taux de communication par robot : 1 message toutes les 3 secondes !),
- Communication par réseau sans fil entre les robots, complétée d'une connexion à un réseau TCP/IP de PC fixes [Kaga et al., 2000], idem pour [Goldberg and Mataric, 1999] (seulement 2.5 bytes/sec/robot).

Notons enfin que ce type d'approche est inadaptée aux systèmes multi-agents évoluant en environnement réel et inconnu, puisqu'il faut garantir que tout agent puisse rester

constamment en contact avec la machine centrale. De plus, par définition, elle ne tient pas compte de l'aspect local des communications que nous utilisons.

### 11.2.2 Solutions distribuées

Les protocoles utilisés dans les réseaux de machines distribuées dépendent de l'existence d'un *accès synchronisé ou non* au médium de communication. Lorsque les stations sont synchronisées, les systèmes utilisent des techniques de multiplexages. Lorsque les stations sont asynchrones, on emploie généralement des variantes du protocole CSMA (cf. ci-dessous), par exemple dans les réseaux locaux de machines (LAN).

Principe du *Multiplexage* : Le canal de communication est partagé de façon égalitaire entre les stations. Il s'agit d'un découpage temporel ou en fréquences du canal, où chaque agent se voit alloué une partie du découpage. Dans le cadre de la division temporelle, les agents peuvent utiliser à tour de rôle la totalité de la bande passante. Cette technique a deux inconvénients majeurs : les stations doivent être *parfaitement synchronisées* et le nombre d'utilisateurs est limité par le découpage fixé initialement (ces techniques sont présentées en détails dans [Tanenbaum, 1998]).

Principe du *CSMA* (Carrier Sense Multiple Access) : Les stations sont indépendantes et par conséquent elles sont en compétition pour l'accès au canal de communication. Ces protocoles écoutent ce qu'il se passe sur le canal avant d'émettre avec une certaine probabilité une trame. Si une collision est détectée, la station attend une durée aléatoire (courte) avant de réitérer le processus d'émission. Ces techniques sont très efficaces tant que le taux de charge du canal reste *inférieur au maximum*. Dans le cas contraire, où toutes les stations ont quelque chose à transmettre, l'efficacité chute fortement (cf. présentations des protocoles à base CSMA dans [Tanenbaum, 1998]).

Cette dernière technique semble adaptée à notre cas de figure. Pourtant, notre modèle nous place dans la situation extrême de forte charge du réseau, car tous les robots peuvent vouloir communiquer simultanément de nombreuses informations (c'est d'autant plus vrai que les robots implémentés utilisent les mêmes signaux infrarouges pour communiquer les valeurs de satisfaction et détecter en continu les obstacles, cf. chapitre 12). Donc, cette approche ne peut répondre complètement à nos exigences d'équité, de forte charge et de robustesse.

Notons qu'en général ce sont des protocoles de type CSMA/CD (Collision Detection)

qui ont été largement implémenté sur des systèmes distribués de robots mobiles autonomes. J. Wang et S. Premvuti ont proposé un protocole faisant référence nommé CSMA/CD-W [Wang et al., 1995].

Actuellement, ces techniques évoluent vers l'utilisation de protocoles internet (TCP/IP) à l'aide de cartes PCMCIA Wireless LAN embarquées sur les robots [Winfield and Holland, 2000]. Le réseau peut ainsi être facilement connecté à un PC fixe extérieur. Cependant, malgré l'utilisation d'émetteurs/récepteurs UHF à fort débit, le taux de communication global reste faible : 10 messages par robot par seconde [Winfield and Holland, 2000].

Malgré l'existence de ces différentes techniques, nous avons entrepris d'élaborer un protocole original adapté aux communications de *nombreux* robots mobiles autonomes. Les principales raisons qui nous ont poussé à cela sont les suivantes :

- besoin d'un système capable de supporter de forte charges du réseau,
- recherche d'une solution "SMA" adaptée à la dynamique du système,
- recherche d'une solution peu coûteuse.

Pour répondre à ces objectifs, notre stratégie a été d'adapter l'approche "multiplexage" à des robots autonomes, donc asynchrones.

## 11.3 Proposition d'un protocole dynamique

### 11.3.1 Spécificités d'un réseau de robots mobiles autonomes

#### Dynamique des réseaux

Les robots considérés dans le problème ont pour particularité d'être très souvent en déplacement dans l'environnement (tâches d'exploration, de transport, etc...). Ils peuvent agir isolément ou travailler à plusieurs sur un objet (manipulation, transport, etc...). Ainsi, leurs distances relatives peuvent rester à peu près constantes, créant des groupes temporaires, mais elles peuvent aussi varier brusquement.

Chaque agent a donc un voisinage de communication très variant (au sens du graphe de communication). En particulier, si la portée des signaux est faible, les agents peuvent alterner souvent communication possible/impossible *tout en restant proches*. Le protocole proposé ci-après tient compte de cette notion d'alternance, en ne perdant pas l'organisation des communications établies dès lors que les agents s'éloignent les uns des autres.

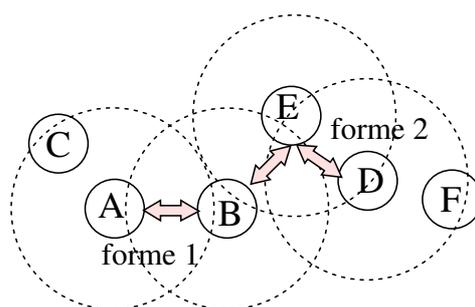


FIG. 11.2 – Les formes de collisions directe (1) et indirecte (2).

Un deuxième aspect fondamental aux communications asynchrones est l'apparition quasi-inévitable d'interférences entre messages, nommées aussi collisions de trames. Dans les systèmes de robots mobiles autonomes, le risque de collisions entre messages n'est pas nul, puisque sans traitement du problème, leur autonomie implique leur asynchronie.

**Définition** (Collision dans un réseau à diffusion [Tanenbaum, 1998]) : Si deux trames sont émises simultanément, les signaux qui les représentent se mélangent et le résultat est inexploitable.

Si deux trames sont émises simultanément et à *proximité*, les signaux qui les représentent peuvent se mélanger. Toutefois, dans un réseau de robots mobiles, la dimension géométrique du problème intervient dans la définition même de ces collisions.

Si l'on considère deux agents A et B suffisamment proches pour communiquer, le problème de *collision* apparaît dès lors que l'agent B transmet une information à l'agent A en même temps que A émet un message vers B. Nous noterons :

**Définition** (*Collision de forme 1* : interférence dans le transport)

Interférence des messages *échangés* simultanément par deux agents sur le même canal.

Le schéma figure 11.2 représente un système de robots mobiles communiquant par diffusion de signaux locaux. La flèche entre les deux robots nommés A et B illustre une collision de forme 1 (directe), c'est-à-dire que les agents sont suffisamment proches pour s'échanger directement des informations.

Toujours dans cet exemple (figure 11.2), les robots B et D sont trop éloignés pour pouvoir communiquer directement. Toutefois, ces deux robots peuvent échanger des informations avec le robot E. Il existe alors un risque de collision à la *réception*, puisque deux messages émis simultanément par B et D peuvent interférer à la réception par E. Dans ce cas nous noterons collision de forme 2, cf. ci après.

**Définition** (*Collision de forme 2* : interférence à la réception)

Réception sur le même canal de plusieurs messages mélangés provenant d'agents ne pouvant communiquer entre eux.

Cette deuxième forme de collision est plus complexe à traiter, car contrairement à la première, seul l'agent recevant les trames mélangées peut s'en apercevoir. En effet, les agents qui la provoquent ne communiquent pas directement entre eux et n'ont pas de retour de l'agent vers qui ils émettent. Nous donnons plus loin une solution robuste et réactive à ce problème.

### Nouvelles contraintes

Le protocole recherché doit posséder les caractéristiques suivantes :

- assurer de nombreuses communications simultanées en garantissant les principes d'équité et de robustesse,
- minimiser les risques de collisions de formes 1 et 2, pouvoir les détecter et les résoudre rapidement,
- assurer un débit de communication régulier malgré les déplacements incessants des agents.

### 11.3.2 Proposition d'un protocole par allocation dynamique et synchronisation indirecte

Le protocole que nous avons développé pour répondre à l'ensemble des contraintes imposées par un réseau formé de robots mobiles autonomes consiste à réaliser *un multiplexage temporel entre stations asynchrones*.

Pour permettre à chaque robot d'émettre quasi-immédiatement une information (principe d'équité), nous avons basé notre solution sur une approche TDMA (time division multiple access). Et plus particulièrement sur le modèle TDMA avec réservation de Crowther [Crowther et al., 1973] qui permet de gérer un nombre inconnu et dynamique d'utilisateurs du canal (principe de robustesse), cf. [Tanenbaum, 1998].

Normalement, ce type de technique nécessite une parfaite synchronisation des stations, ce qui n'est pas le cas des robots envisagés. Nous avons alors développé une forme de synchronisation permettant d'implémenter l'approche TDM entre stations *asynchrones* (et non nécessairement capables d'écouter pendant une émission).

La synchronisation des robots peut être envisagée par l'implémentation d'un processus obligeant tous les agents à préalablement se synchroniser pour pouvoir communiquer,

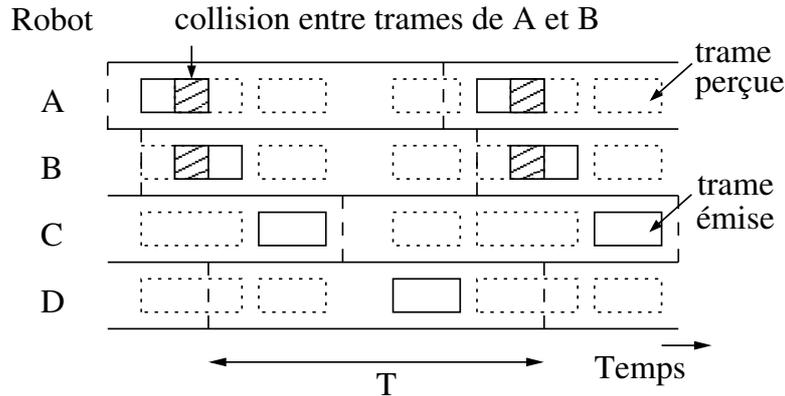


FIG. 11.3 – Protocole proposé : synchronisation indirecte des agents par cycles de durée commune  $T$  et allocation d'une zone d'émission par chaque agent sur son cycle après écoute.

menant ainsi le système à une synchronisation globale. Mais cette solution pose divers problèmes : (i) tout processus de synchronisation est coûteux en temps, ce qui n'est pas compatible avec notre objectif de réactivité, (ii) cette solution peut mener à la synchronisation des agents d'un groupe (ou agents voisins), mais pas à celle de plusieurs groupes lorsqu'ils se rencontrent.

Une autre approche proposée dans [Elson and Estrin, 2001] consiste à utiliser une balise qui émet une pulsation régulière à partir de laquelle les robots peuvent se synchroniser. Mais comme le précisent les auteurs, cette solution est mal adaptée aux environnements étendus. De plus, comme tout système centralisé, cette solution reste fragile.

---

**Proposition :** Notre solution emploie une forme de *synchronisation implicite*. Le principe général repose sur le fait que tous les agents entretiennent *un cycle temporel de même durée constante  $T$* . Chaque agent peut alors émettre des trames sur une portion fixe de son cycle, après s'être assuré qu'elle paraissait libre. Ainsi, les autres agents perçoivent cette réservation comme un intervalle statique sur leur propre période, puisqu'ils gèrent eux aussi un cycle de durée  $T$ .

Si l'on considère seulement deux agents, leurs cycles sont *déphasés* d'une durée *constante*. Ce déphasage constant permet à chaque agent de repérer très précisément les zones libres et occupées de la période  $T$  (cf. principe figure 11.3). En cas de collision directe entre deux agents, il suffit que l'un d'eux cherche un nouvel "emplacement" sur les portions libres de la période.

---

### Division temporelle du canal pour une allocation dynamique asynchrone

Conditions technologiques sur le système physique :

- chaque agent, muni d'une horloge précise, doit gérer un cycle temporel (ou période) d'une durée constante notée  $T$  (cf. fig. 11.3),
- deux cas d'études : (i) les agents peuvent écouter le canal à tout moment (ii) uniquement lorsqu'ils n'émettent pas.

**Protocole d'allocation** pour chaque agent (basé sur une écoute de la porteuse Carrier Sense) :

Avec écoute possible pendant l'émission :

1. avant d'émettre, l'agent écoute le canal (pendant au moins une période  $T$ ),
2. sur les zones libres, l'agent tire aléatoirement une position  $t_e$  (si pas de zone libre de taille supérieure à  $T_e$ , réitérer l'étape 1).
3. l'agent émet une trame de durée  $T_e$  à l'instant  $t_e$
4. si l'agent détecte une collision, il fait une réallocation avec une probabilité  $p$ , puis retourne à l'étape 1.
5. tant que l'agent ne détecte pas de collision, il utilise régulièrement l'intervalle  $[t_e, t_e + T_e]$  pour émettre : étape 3.

Le principe de synchronisation implicite permet d'utiliser, entre stations indépendantes, une approche par "réservation". Par contre, ces réservations sont gérées dynamiquement par une écoute de la porteuse.

Nous introduisons plus loin une petite marge de sécurité temporelle séparant en pratique les zones d'allocations.

Nous avons aussi envisagé l'implémentation de ce protocole dans le cas où les robots seraient incapables d'écouter la porteuse pendant une émission.

Sans écoute possible pendant l'émission :

1. avant d'émettre, l'agent écoute le canal (pendant au moins une période  $T$ ),
2. sur les zones libres, l'agent tire aléatoirement une position  $t_e$  (si pas de zone libre de taille supérieure à  $T_e$ , réitérer l'étape 1).
3. l'agent émet avec une probabilité  $q$  une trame de durée  $T_e$  à l'instant  $t_e$

4. si l'agent n'a pas émis (probabilité  $1 - q$ ), il écoute le canal pendant toute la durée  $T$ , s'il y a réception d'informations à  $[t_e, t_e + T_e]$  (donc collision) alors il fait une réallocation avec une probabilité  $p$ , puis retourne à l'étape 1.
5. tant que l'agent ne détecte pas de collision, il utilise régulièrement l'intervalle  $[t_e, t_e + T_e]$  pour émettre : étape 3.

Cette variation introduit une probabilité  $q$  préfixée d'émission des trames. Si celle-ci n'a pas lieu, l'agent écoute sa zone d'émission pour vérifier qu'elle n'est pas utilisée par un autre agent. Si la zone n'est pas libre, c'est qu'il y a collision.

### Détection des collisions de forme 2 :

Le protocole d'allocation proposé est capable de détecter les collisions de forme 1 (échanges simultanés d'informations entre au moins deux robots voisins, ligne 4). Cette détection peut avoir lieu à chaque intervalle (ou avec une probabilité  $1 - q$  si l'agent ne peut écouter pendant l'émission).

Si l'on souhaite détecter les collisions de forme 2 (interférences à la réception), il faut ajouter à la *réception* (intervalle  $[0, t_e] \cup [t_e + T_e, T]$ ) l'opération suivante :

Pour toute trame reçue à un temps  $t_c$ , si elle n'est pas cohérente (détection d'une interférence) la rejeter et émettre régulièrement sur l'intervalle  $[t_c, t_c + T_e]$  un signal de *brouillage*, jusqu'à ne plus recevoir de trames altérées.

Le principe est simple : tout agent qui détecte une collision de forme 2 (interférence à la réception) émet un signal de brouillage dans le but de forcer une réallocation des agents provoquant la collision.

Dans le cas d'un système sans écoute possible pendant l'émission, ce principe fonctionne en introduisant une probabilité  $p'$  d'émission des trames de brouillage. Ainsi, lorsque l'agent n'émet pas de trame, il écoute la zone d'émission pour vérifier la fin de la collision, sinon il continue le brouillage.

### 11.3.3 Résolution des collisions et convergence

Nous supposons que  $T$  est suffisamment grand devant  $T_e$  pour assurer une allocation possible à tout agent à chaque itération du cycle. Notons que dans le cas contraire, le protocole fait patienter l'agent jusqu'à libération d'une intervalle  $T_e$  (ligne 2).

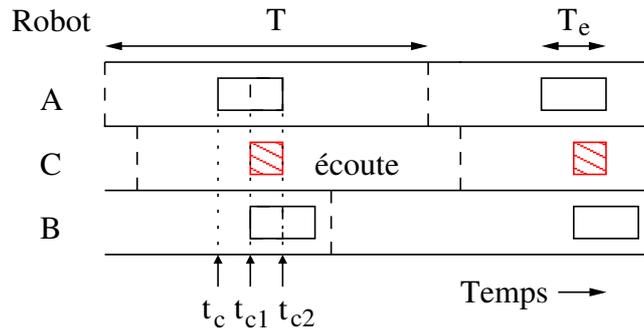


FIG. 11.4 – Collision à la réception par C de deux messages provenant de A et B

### Principe d'émission par réservation dynamique :

Nous avons vu qu'une forme de synchronisation existait entre les agents, par le fait qu'ils géraient un cycle temporel de même durée. Il leur est ainsi possible de détecter de façon statique les zones libres de leur cycle. Le principe d'allocation/réservation d'une portion de la période devient alors équivalent au modèle TDMA avec réservation de Crowther [Crowther et al., 1973]. Nous examinons maintenant la résolution des deux formes de collisions.

### Gestion des collisions

Situation de collision : plusieurs agents voisins décident d'émettre en même temps (le canal est libre pour tous), certaines zones d'allocations ont des intersections non nulles.

Considérons une collision de forme 1 entre  $n$  agents  $a_1, \dots, a_n$ . Les  $n - 1$  premiers agents qui détectent la collision réallouent leur intervalle d'émission dans des zones libres (ligne 4 du protocole). Ainsi, ces agents ne sont plus en collision avec le  $n^{ieme}$  agent (probabilité  $\simeq 1-p^2$ ), qui se trouve alors libéré de la collision initiale et conserve sa position d'émission.

Les  $n - 1$  nouvelles allocations peuvent provoquer de nouvelles collisions puisque certaines peuvent avoir lieu en même temps. Dans ce cas, le processus décrit précédemment agit aussi longtemps qu'il y a des collisions. Le processus termine puisque nous avons l'hypothèse que tous les agents peuvent trouver un intervalle libre  $T_e$  à chaque période  $T$ .

Remarque : une collision entre deux agents ne nécessite qu'une seule réallocation.

Collision de forme 2 :

Considérons deux agents  $A$  et  $B$  suffisamment distant pour ne pas pouvoir communiquer directement entre eux et qui émettent des messages vers un troisième agent  $C$  (cf. fig. 11.4).

A la réception des messages,  $C$  perçoit une collision de forme 2. Supposons que  $A$  émette sa trame légèrement avant  $B$ , alors la collision des messages a lieu pour  $C$  sur un intervalle noté  $[t_{c1}, t_{c2}]$ , où  $t_{c1}$  est le début de réception de la trame de  $B$  et  $t_{c2}$  la fin d'émission de la trame de  $A$ . Notons, toujours dans le référentiel de  $C$ , le début de réception de la trame de  $A$  au temps  $t_c$  ( $t_c < t_{c1}$ ).

Lorsque l'agent  $C$  détecte l'interférence il émet un signal de brouillage sur l'intervalle  $[t_c, t_c + T_e]$ . Donc, l'agent qui a transmis la première trame concernée par la collision ( $A$ ) va recevoir ce brouillage exactement sur son intervalle d'émission.

Nous avons  $t_{c1} < t_c + T_e$ , alors le message de brouillage émis à  $t_c$  va être capté par le deuxième agent ( $B$ ) sur la période  $[t_{c1}, t_{c2}]$ . Il provoque ainsi une collision volontaire avec le début d'émission des trames de  $B$ .

Les deux agents  $A$  et  $B$  sont donc "touchés" par la trame de brouillage, sous formes de collisions de forme 1. Le premier des deux agents qui change sa zone d'émission annule la collision de forme 2. Aussitôt, l'agent  $C$  ne reçoit plus de trames altérées et arrête ses émissions.

Ce principe s'applique à l'interférence d'un nombre quelconque d'agents, puisque l'agent détectant une collision de forme 2 émet un brouillage tant que les trames reçues sont altérées. Les agents impliqués perçoivent tous une collision de forme 1, or nous avons vu précédemment que cela entraînait la réallocation des agents concernés vers des zones libres.

Ces processus fonctionnent de la même manière avec le protocole sans écoute de la porteuse, puisque l'émission des trames n'est modifiée que par une probabilité  $1 - q$  d'écoute pour détecter les collisions de forme 1. Par la suite nous ne considérerons plus cette variante, mais les résultats lui restent applicables.

## 11.3.4 Paramètres et efficacité

### 11.3.4.1 Stabilité face à la mobilité des agents

Nous avons évoqué en section 11.3.1 les problèmes de communications liés à la dynamique d'un système de robots mobiles. En particulier aux risques de coupures (déconnexions) incessantes dues à la mobilité des agents, et ce même s'ils appartiennent au même groupe.

Le protocole que nous venons de décrire résiste bien à ce problème. En effet, lorsque les agents réservent une zone d'émission, celle-ci reste la même tant qu'il n'y a pas perception d'une interférence. Ainsi, des agents agissant à proximité ou en groupe peuvent

converger vers un état stable (sans collisions) de leur zones d'émissions et le conserver. Si certains agents s'éloignent temporairement alors toute reprise de communication n'entraînera aucune nouvelle interférence puisque chaque agent aura conservé sa même zone d'émission.

Enfin, si les agents n'émettent pas en continu, toute nouvelle émission de trames (donc allocation) est précédée d'une écoute du canal pendant une durée  $T$ , minimisant les risques de collisions. Il en est de même d'un nouveau robot qui souhaiterait "se joindre" aux communications d'un groupe.

#### 11.3.4.2 Fréquence fixe pour une "infinité" d'agents communicants

Les paramètres fondamentaux du protocole sont la période  $T$ , la probabilité  $p$  de réallocation en cas de collision de forme 1 et la fonction qui détermine la position d'allocation. Notons que la durée des trames  $T_e$  est fixée par le concepteur/utilisateur.

Le premier paramètre dépend directement du nombre d'agents pouvant communiquer simultanément ensemble. La particularité des systèmes que nous considérons est de diffuser *localement* les informations. Ainsi, **chaque agent a un nombre maximum de voisins pouvant être simultanément en communication avec lui.**

Les agents ayant tous la même portée d'émission ce voisinage maximum est commun (noté  $N_c$ ) et il dépend directement du rayon d'émission et de la taille des agents (formule du type  $\frac{\text{aire du signal}}{\text{aire d'un agent}}$ ).

Pour garantir le principe d'équité et éviter les famines, il faut vérifier l'hypothèse de l'existence d'une période  $T$  suffisamment grande pour que chaque agent puisse réserver une durée  $T_e$  sans collisions.

Considérons une période  $T$  donnée et supposons que l'allocation se fasse aléatoirement mais en évitant les collisions. Dans le pire des cas, les trames seront espacées entre elles d'un intervalle  $T_e - \varepsilon$ , où  $\varepsilon$  est très petit. Ainsi, aucune trame ne peut être insérée dans les intervalles libres. Nous en déduisons que, dans le pire des cas, chaque allocation peut réduire l'espace libre d'une durée de  $2.T_e - \varepsilon$ . Donc, pour s'assurer que chaque agent puisse trouver une zone libre, il faut choisir  $T$  tel que  $T \geq 2.N_c.T_e$ .

Ce résultat est intéressant car, une fois fixé  $T$  vérifiant cette inégalité, quel que soit le *nombre* d'agents qui *entrent* ou sortent du système, chacun d'eux est assuré de pouvoir toujours communiquer avec l'ensemble de ses voisins.

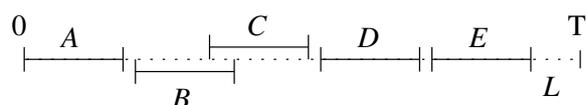


FIG. 11.5 – Cas extrême de collision (point de vue général)

### 11.3.4.3 Optimisation de la fonction d'allocation et de la période $T$

Nous avons précisé qu'une (ré)allocation se faisait par un choix de position *aléatoire* dans une zone libre. Mais cette stratégie simpliste nous mène à définir une période minimale de durée double du minimum suffisant théorique  $N_c.(T_e + \epsilon)$  ( $\epsilon$  est la durée minimale séparant deux zones).

Nous traitons ici le problème de la réduction de la période  $T$  pour augmenter le taux de transfert d'informations par unité de temps. Pour cela nous avons déterminé une fonction d'allocation minimisant les écarts entre zones occupées. L'objectif étant de tendre vers une période  $T$  de durée optimale, c'est-à-dire minimale, de valeur  $N_c.(T_e + \epsilon)$ .

L'idée que nous avons appliquée consiste à "coller" les zones d'allocations pour ne pas créer de zones non allouables, c'est-à-dire de s'éloigner au maximum du pire des cas évoqué précédemment.

#### Choix de la zone de (ré)allocation :

Lorsqu'un agent doit choisir une zone d'émission, il connaît sur sa période  $T$  toutes les zones libres de durée supérieure ou égale à  $T_e + 2.\epsilon$ . L'agent ne s'intéresse qu'à ces zones car elles peuvent contenir la durée d'une trame :  $\epsilon + T_e + \epsilon$ .

La position d'émission  $t_e$  est calculer en utilisant un tirage aléatoire sur les zones libres :

S'il existe au moins une zone libre supérieure ou égale à  $T_e + 2.\epsilon$  alors le nouveau temps d'émission est tiré aléatoirement suivant une distribution uniforme parmi les zones libres. Puis dans la zone choisie,  $t_e$  est "collée" à droite ou à gauche de la zone (et à une distance  $\epsilon$ ).

L'opération de concaténation des zones occupées permet de ne pas créer de zones libres de durée inférieure à  $T_e + 2.\epsilon$ , qui sont inutilisables pour d'autres allocations.

Toutefois, cette technique ne permet pas de garantir à elle seule qu'un nombre  $N_c$  d'agents trouvent une solution sans collisions dans une période de durée  $N_c.(T_e + \epsilon)$ .

Par exemple la figure 11.5 présente la superposition des perceptions d'un système de cinq agents qui tentent tous d'allouer une zone pour communiquer. Malgré l'espace théo-

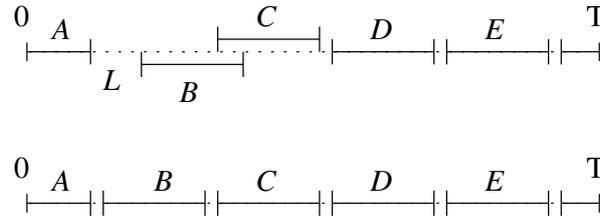


FIG. 11.6 – Glissement des allocations de A puis de B

riquement suffisant, la collision entre B et C ne peut être résolue par l'application de la fonction proposée ci-dessus.

En effet, les agents A, D et E ne perçoivent pas de collision, donc ils ne cherchent pas à ré-allouer leur zone d'émission. Par contre, B et C sont en collision. Dans ce cas, B et C peuvent tenter de trouver une nouvelle zone d'émission dans l'espace libre, mais ici cet espace (noté  $L$ ) est très inférieur à  $T_e + 2\epsilon$ .

En conclusion, pour que ce problème trouve une solution, il faut que ce soit les autres agents qui modifient leur position d'allocation. Toutefois, il faut établir un mécanisme modifiant le moins possible les solutions déjà stables du système.

### Opération de glissement

Au protocole proposé nous ajoutons à chaque itération du cycle T :

Si l'agent ne vient pas de modifier sa zone d'émission et s'il existe un espace libre  $e$  qui précède immédiatement sa position d'émission  $t_e$ , et

Si  $e > \epsilon$  et  $e < T_e + 2\epsilon$  alors modifier  $t_e$  en lui retranchant la durée  $e$  :

$$t_e \leftarrow (t_e - e) \bmod T$$

Si un glissement à lieu, il ne peut pas créer de nouvelle collision, puisque le glissement se fait vers une zone "libre" (exemple : glissement de A en fig. 11.6). Par contre il peut entraîner l'élimination d'une collision (exemple : glissement de B en fig. 11.6).

Dans le système figure 11.5, la convergence vers une solution est bien réalisée par les étapes de glissements : d'abord de A (c'est le seul à avoir un espace de glissement), puis de B (dans l'espace créé par le glissement de A).

L'application systématique de cette opération conduit à éliminer tout espace libre non allouable de la période. Par conséquent, les allocations convergent vers une situation de concaténation totale, c'est-à-dire la situation optimale :  $N_c$  zones allouées sur une période de durée  $N_c.(T_e + \epsilon)$ .

Remarque importante : Si  $N_c.(T_e + \epsilon) < T < (N_c + 1).(T_e + \epsilon)$  alors le processus de glissement risque de s'exécuter indéfiniment. En effet, le "collage" de tous les  $T_e + \epsilon$  laisserait toujours un espace  $e$  compris entre  $\epsilon$  et  $T_e + 2.\epsilon$  qui entraînerait un nouveau glissement. Pour éviter une telle situation, il suffit de choisir  $T$  comme un multiple de  $T_e + \epsilon$  ou simplement supérieur à  $(N_c + 1).(T_e + \epsilon)$ .

#### 11.3.4.4 A propos de l'implémentation du protocole

Les trames étant composées de bits, nous prendrons la durée d'émission d'un bit comme unité temporelle pour exprimer  $T$  et  $T_e$ . Nous précisons ici les structures informatiques nécessaires à la programmation d'un tel protocole :

- nécessité de stocker un vecteur de bits de longueur  $T$  (au plus quelques centaines, cf. chapitre 12), pour connaître l'état d'occupation du cycle ainsi que les messages reçus.
- le cycle temporel est réalisé par une simple boucle, dont la période est gérée par appel direct à l'horloge des machines,
- pour optimiser le calcul de la fonction d'allocation et l'opération de glissement, il est nécessaire de stocker pendant le balayage d'un cycle les zones occupées sous formes d'intervalles.

Les opérations de stockage, du vecteur de  $T$  bits et des intervalles d'occupations sont de très faible coût mémoire puisque à chaque itération les anciennes informations sont écrasées. Le protocole ne mémorise rien d'autre que le dernier cycle qui vient de s'écouler. Nous revenons en détails sur l'implémentation en C de ce protocole dans le chapitre suivant.

## 11.4 Simulations et optimisation de la probabilité de réallocation

En vue d'implémenter réellement le protocole sur des robots mobiles, nous l'avons préalablement évalué à l'aide d'un outil de simulation. Nous avons développé une application en Java permettant d'observer les processus d'allocations, de résolution des collisions et de glissement des zones d'émissions. Notre but était aussi d'évaluer les temps de convergence en cas de collisions et d'étudier l'influence de la probabilité  $p$  sur ces durées. Le fonctionnement de ce simulateur ainsi que la représentation graphique qu'il donne des différents processus du protocole sont donnés en Annexe 14.3.

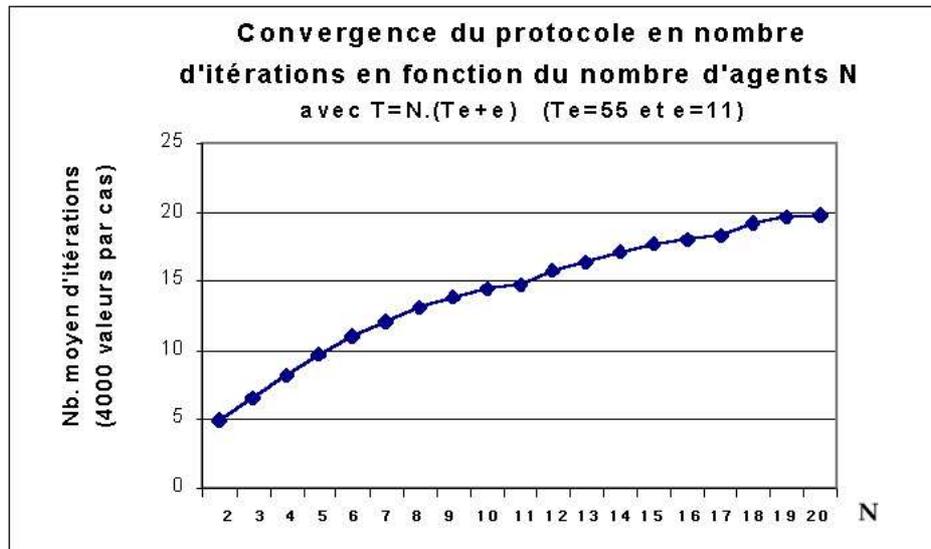


FIG. 11.7 –

## Validation

Au delà de l'aspect "observation" du fonctionnement du protocole, le simulateur nous a permis d'évaluer ses performances pour des situations extrêmes. Nous simulons toujours  $N$  agents proches tentant de communiquer tous simultanément. Une telle situation crée naturellement de nombreuses collisions. Le simulateur nous permet alors d'observer la résolution et de mesurer sa durée (temps de convergence). Les simulations menées ont consisté à faire varier le nombre d'agents de 2 à 20 et le rapport  $\frac{T}{N_c.(T_e+\epsilon)}$  de 1 à 2.

Pour toutes les simulations effectuées ( $T > N_c.(T_e+\epsilon)$ ), quelque soit le nombre d'agents, le système a toujours convergé. Nous avons aussi vérifié que plus la valeur de  $T$  était supérieure à  $N_c.(T_e+\epsilon)$ , plus rapide était la convergence vers une solution sans collisions.

Nous nous sommes alors intéressé à ces vitesses de convergence pour des systèmes ayant une période quasi-optimale, c'est-à-dire de durée  $N_c.(T_e+\epsilon)$ . La courbe figure 11.7 donne les temps moyens de convergence en fonction du nombre d'agents considérés ( $T_e = 55$  et  $\epsilon = 11$ ). Il est intéressant de constater que malgré l'augmentation du nombre d'agents (de 2 à 20), donc du nombre de collisions potentielles (en  $O(n^2)$ ), le nombre d'itérations nécessaire à la convergence reste linéaire (le traitement se fait bien en parallèle).

## Étude du paramètre $p$

Le paramètre  $p$  représente la probabilité qu'un agent exécute une réallocation lorsqu'il détecte une collision directe avec au moins l'un de ses voisins. Initialement, nous avons

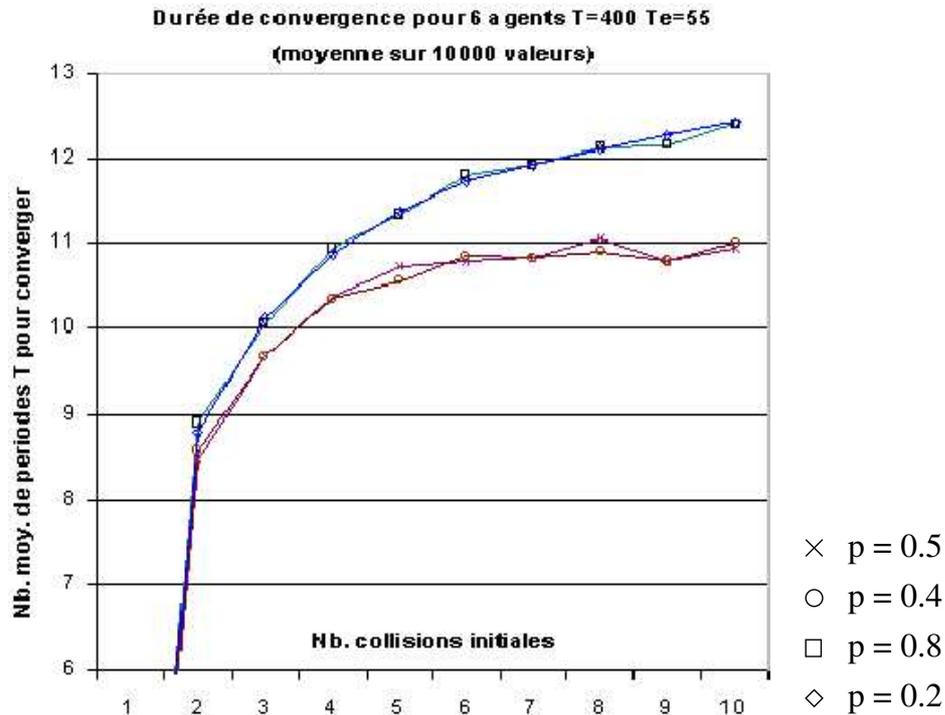


FIG. 11.8 – Étude de la durée de convergence du protocole d'allocation pour différentes valeurs de  $p$  en fonction du nombre de collisions initiales.

posé  $p = 0.5$ . Ce choix s'appuie sur le fait qu'une collision entre deux agents ne nécessite la réallocation que de l'un d'entre eux. De plus, si dès la première détection, les deux agents exécutent une réallocation (cas  $p = 1$ ), ils augmentent leur risque d'engendrer de nouvelles collisions ou de ne pas trouver de zones libres.

Toutefois, une collision directe peut avoir lieu entre plus de deux agents à la fois, ou inversement, une réallocation peut résoudre simultanément deux collisions. Ces situations particulières nous ont poussé à essayer de mieux comprendre l'influence du paramètre  $p$  sur les temps de convergence, ainsi que l'effet du nombre de collisions initiales sur ces temps.

Pour cela, nous avons effectués des simulations d'une situation référence, en faisant varier le paramètre  $p$  et en "triant" les résultats suivant le nombre de collisions initiales.

Le système de référence est composé de six agents,  $T_e = 55$  et  $T = 400$  (soit  $\epsilon = 11$ ). Les courbes tracées donnent en ordonnée le temps moyen de convergence et en abscisse le nombre de collisions initiales. Nous avons limité l'axe des abscisses à 11 collisions initiales, car au delà les situations étaient trop rarement obtenues. Chaque point est calculé en faisant la moyenne de 10000 résultats. Nous donnons en figure 11.8 les courbes obtenues pour quatre valeurs de  $p$  :

1.  $p = 0.5$  , probabilité moyenne, les résultats sont quasi-optimaux,
2.  $p = 0.4$  , meilleurs résultats obtenus,
3.  $p = 0.8$  , probabilité très forte, les réallocations sont trop nombreuses, les résultats sont en moyenne moins bons, jusqu'à 1,5 itérations en plus pour 11 collisions initiales,
4.  $p = 0.2$  , probabilité très faible, les réallocations ne sont pas effectuées assez rapidement, les résultats sont aussi peu performants que pour une trop forte probabilité.

Ces tracés confirment que des probabilités extrêmes (ici 0.8 et 0.2) donnent des temps moyens très éloignés des meilleurs résultats. Par contre, la valeur 0.5 initialement proposée est quasi-optimale (justifiant son emploi lors de l'implémentation réelle sur des robots, cf. chapitre suivant).

Il est intéressant de noter que les courbes tendent rapidement vers des maxima de moyenne inférieure à 13 itérations. Ceci s'explique par le fait qu'une situation initiale avec de nombreuses collisions est due à l'allocation par les agents de la même zone temporelle. Il existe donc de grandes zones libres qui sont rapidement utilisées pour résoudre ces collisions.

## 11.5 Bilan sur le protocole proposé

Nous reprenons les aspects originaux du protocole proposé :

- il est complètement distribué et n'utilise pas de *processus* de synchronisation,
- il implémente une forme de multiplexage temporel entre stations asynchrones, donc il assure un débit constant d'informations échangées, même en cas de charge maximale du réseau,
- la fréquence de communication ( $1/T$ ), proportionnelle à la portée des signaux, est indépendante du nombre d'agents dans le système et assure le principe d'équité à tous les agents (égalité de chance pour accéder au médium de communication),
- la mobilité des agents qui peut provoquer de nombreuses déconnexions temporaires ne remet pas en cause la stabilité des allocations établies et ne crée pas de nouvelles collisions (idem pour les pannes),
- l'implémentation est simple et peu coûteuse en espace mémoire, donc adaptée aux architectures réactives.

Ce protocole a été implémenté et validé sur des robot mobiles réels (cf. chapitre suivant). La fréquence de communication dépend de la durée de la période  $T$ . Celle-ci étant exprimée en bits, il nous faut préciser la fréquence d'émission des bits avant de parler de fréquence de communication.

Les robots que nous avons développés communiquent par émission de signaux infrarouges (ils ont un démodulateur intégré imposant une fréquence de modulation de 40kHz). Nous avons défini une fréquence d'émission de bits de 1600 (bauds). Ainsi, les fréquences de communications obtenues sont (avec des trames contenant 1 octet) : 30 Hz pour 2 robots (30 octets/sec/robot), 22 Hz pour 3 robots et (théorique) 8Hz pour 10 robots (cf. chapitre suivant). Ces valeurs sont supérieures aux fréquences de communication de [Goldberg and Mataric, 1999] et [Parker, 1999].

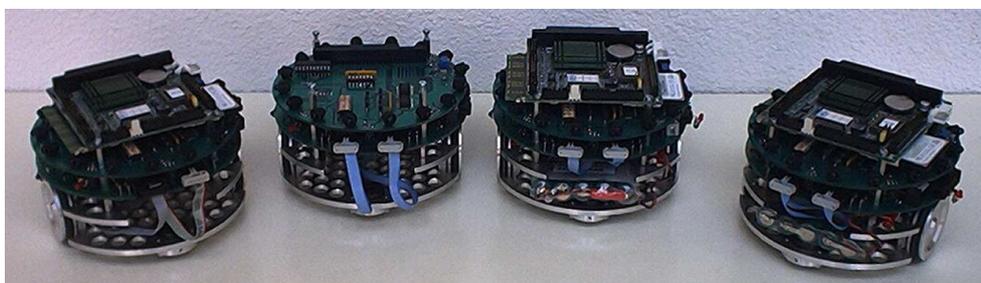
Remarquons que si nous avions employé une porteuse de fréquence plus élevée, comme celles d'émetteurs HF, de l'ordre de 400MHz, alors nous aurions eu un taux de transfert de 4800 bauds (type Khepera). Dans ce cas, notre protocole fournirait une fréquence de communication de 24 Hz pour 10 agents (89 Hz pour 2 agents).

Le chapitre suivant présente l'implémentation de ce protocole sur des robots mobiles autonomes ainsi que des expérimentations de résolutions distribuées de conflits spatiaux.



# Chapitre 12

## Validation sur des mini-robots mobiles



### 12.1 Introduction

Une collaboration fructueuse s'est établie au LIRMM<sup>1</sup> tout au long de cette thèse entre les équipes des professeurs Alain Liégeois (robotique) et Jacques Ferber (informatique).

En début de thèse, notre ambition était essentiellement l'élaboration et la simulation d'un modèle d'agents coopératifs. La validation sur des robots réels n'étant alors qu'un objectif flou et incertain, mais cependant présent tout au long de notre étude.

En octobre 2000, Philippe Lucidarme a débuté une thèse de robotique, dirigée par Alain Liégeois, traitant des problèmes d'apprentissage dans les systèmes multi-robots mobiles. Mais par cette même occasion, nous avons eu la chance d'accueillir un formidable concepteur de robots mobiles (cf. [Lucidarme, 2001]). Rapidement, une collaboration entre les deux thèses s'est mise en place pour aboutir au développement d'un système de robots mobiles autonomes communicants.

P. Lucidarme a conçu un premier prototype sur lequel j'ai programmé un module (ou

---

<sup>1</sup>Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier

noyau) de navigation réactive, dont les caractéristiques sont données plus loin. J’ai ensuite implémenté le protocole de communication présenté au chapitre 11, basé sur l’émission de signaux infrarouges, qui nous a permis à la fois de faire communiquer les robots et d’assurer la détection des obstacles. Nous avons alors à notre disposition un véritable système multi-agents robotisé “prêts à l’emploi” (cf. photo en début de chapitre).

A ce stade, j’ai pu programmer le modèle satisfaction-altruisme que nous avons validé par différentes expériences. Nos expérimentations concernent à ce jour essentiellement des problèmes de navigation collective et la résolution des conflits spatiaux (présentés au chapitre 9). Nous avons reproduit un corridor à l’intérieur duquel les robots peuvent se déplacer mais ne peuvent pas se croiser. Ainsi, nous avons pu tester et valider notre architecture, ainsi que l’algorithme distribué de résolution des blocages dans ce type d’environnement.

## 12.2 Navigation autonome et comportements réactifs

Le module de navigation, implémentant le modèle proposé au chapitre 6, est directement couplé aux perceptions de l’agent. Il nous faut donc, dans un premier temps, préciser les caractéristiques techniques des robots utilisés.

### Pourquoi développer nos propres robots ?

L’expérimentation du modèle aurait pu être envisagée par l’utilisation (après achat) de mini-robots mobiles existants. En particulier, par l’emploi de robots Khepera [Khepera, ], convenant bien au “type” de système étudié. Cependant ces derniers nous auraient contraint par une faible puissance de calcul, de mémoire et surtout par un mode de communication très différent de nos besoins (cf. ci-après)

Ainsi, nous avons opté pour le développement, au sein de l’équipe, de robots à la fois adaptés à nos objectifs mais aussi ouverts à de nombreuses autres applications [Lucidarme et al., 2001].

Le prototype proposé est construit “autour” de la carte mère embarquée PC 104 (Advantec), qui nous permet de disposer de

- la puissance d’un véritable PC (486 DX2 66) (et non d’un micro-contrôleur comme sur les Khepera par exemple),
- d’une mémoire vive importante pour un mini robot (8 Mo),
- d’un disque dur flash (amovible) de 8 Mo ou 16 Mo.

L'autre motivation majeure, nous ayant poussé au développement local, tient à la liberté offerte d'implémenter un système de communication peu coûteux et original. L'idée initiale étant d'effectuer les communications seulement par signaux infrarouges utilisés normalement pour la détection d'obstacles. Couramment, les systèmes multi-robots communiquent par l'emploi de modules HF qui sont plus coûteux, plus fragiles et diffusent dans tout l'environnement (pas de notion de localité des communications). De plus, ils nécessitent souvent d'être contrôlés/supervisés par un PC extérieur au système (cf. section 11.2).

Nous avons proposé au chapitre précédent une solution alternative, dont nous présentons ci-après l'implémentation et la validation sur des robots mobiles autonomes.

Le troisième intérêt à notre "autonomie" se situe au niveau des performances mécaniques du robot. Celui-ci peut se déplacer, à l'aide de deux roues indépendants entraînés par des moteurs analogiques, à une vitesse supérieure à 1m/s. D'un poids d'environ 1 Kg, sa vitesse de réaction à des perceptions d'obstacles est rapide. De plus, il peut accueillir facilement des modules supplémentaires, comme par exemple un bras articulé (réalisation d'un bras à 3 degrés de liberté conçu au LIRMM par A. Liégeois, selon les principes d'une architecture mécanique parallèle [Gaudry et al., 1995]).

Enfin, toutes ces caractéristiques sont obtenues pour un coût financier relativement bas (environ 1000 euros), en rapport aux performances offertes et surtout à la flexibilité de contrôle et d'évolution du robot.

### Aspect général des robots

Chaque robot a la forme générale d'un cylindre de hauteur 10 cm et de diamètre 13 cm. Le corps des robots est composé d'un châssis maintenant deux roues indépendantes (cf fig. 12.1.b). Sur cette base s'empilent d'abord une carte d'interface avec les moteurs et les codeurs magnétiques<sup>2</sup>, puis une carte d'interface avec les émetteurs et récepteurs de signaux infra rouge (cf. fig. 12.1.c). Enfin la carte mère du PC embarqué domine l'ensemble (cf. fig. 12.1.d).

Les communications entre les deux cartes sont réalisées par le bus du PC 104. Les 16 émetteurs infrarouge sont répartis à la périphérie supérieure de la première carte interface, les 8 capteurs sont placés à la périphérie inférieure (cf. fig. 12.1.c). La fréquence de modulation des signaux est de 40kHz.

Les batteries, non représentées ici sont placées dans le châssis. Ce sont deux batteries

---

<sup>2</sup>Codeurs magnétiques 16 imp./tour de moteur pour la lecture de la vitesse de la roue (réducteur 1/22).

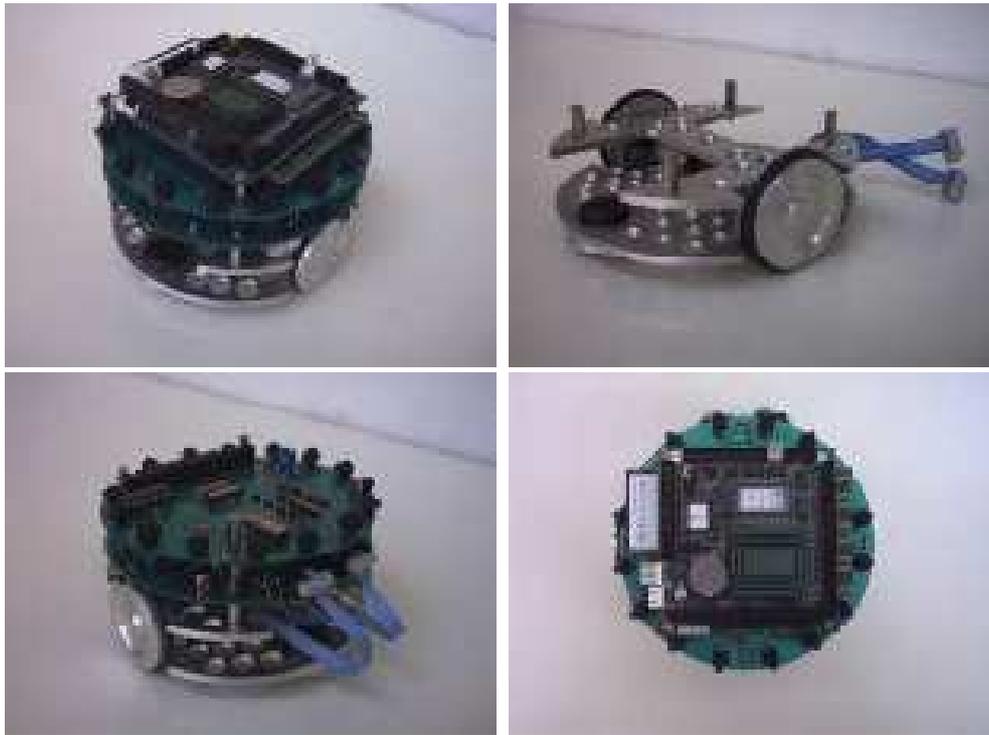


FIG. 12.1 – Prototype d'un mini-robot mobile autonome

Ni-MH assurant plus d'une demi heure d'autonomie. Pour de plus amples informations concernant les caractéristiques et la conception de ces robots, voir [Lucidarme et al., 2002] ou consulter [Lucidarme, 2001].

### 12.2.1 Implémentation des communications et des perceptions

L'intégration du protocole de communication proposé s'est faite par la transposition quasi directe des fonctions développées dans le simulateur du protocole (cf. Annexe 14.3). Nous avons installé sur les robots le système d'exploitation MSDOS pour encombrer le moins possible le disque dur et disposer d'un système sans tâche de fond. Les programmes ont été développés en Borland C++.

Rappelons que le module de communication peut être implémenté comme un processus parallèle au noyau de comportement (navigation et sélection d'actions). Mais l'aspect réactif de notre modèle nous a permis d'intégrer le modèle satisfaction altruisme directement dans la boucle de communication-perception. L'algorithme donné en figure 12.2 résume le fonctionnement général de cette boucle.

Nous précisons ci-dessous l'implémentation informatique et électronique des modules

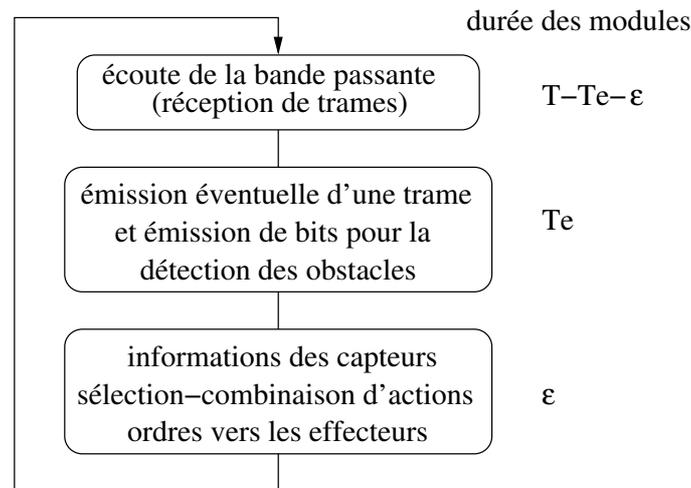


FIG. 12.2 – Algorithme général du cycle communication-perception-action de l'agent

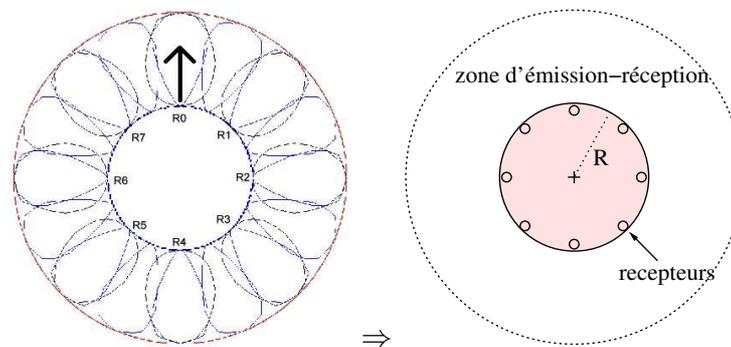


FIG. 12.3 – Diffusion omnidirectionnelle de signaux infrarouges par un ensemble de 16 émetteurs infrarouges. Réception par 8 capteurs infrarouges.

de l'algorithme.

### Perception et communications par signaux IR

Pour réaliser la diffusion locale des informations (signaux) nous utilisons un ensemble d'émetteurs *infrarouges* constituant autour de chaque robot un disque de propagation de rayon paramétrable (principe figure 12.3).

Il est aisé de transmettre une simple pulsation à l'aide d'un émetteur infrarouge. C'est une onde qui se propage face à l'émetteur et qui s'atténue avec la distance. Mais dans notre modèle, il s'agit de transmettre une valeur *signée*, qui ne soit pas nécessairement amortie avec la distance.

Nous avons proposé au chapitre précédent un protocole permettant à chaque agent d'émettre, suivant un taux de transfert fixe, de courtes trames sans accusé de réception.

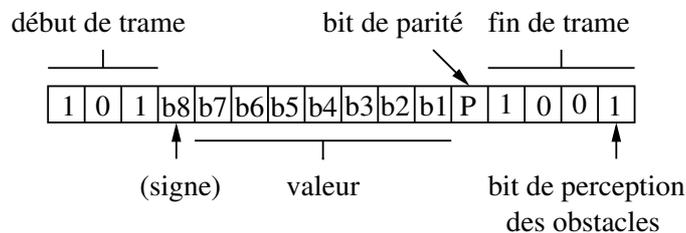
Nous utilisons l'émission de signaux infrarouges pour former ces trames.

Pour appliquer le protocole proposé, il nous suffit de créer une fréquence d'émission de bits sur la fréquence de modulation des signaux infrarouges. Celle-ci est définie simplement par :

- bit de valeur 1 : émission de signaux IR pendant une très courte durée  $T_b$ ,
- bit de valeur 0 : pas d'émission pendant la période  $T_b$ .

Ainsi, nous avons une fréquence d'émission/réception de bits de  $\frac{1}{T_b}$ .

Les trames émises sont courtes : un mot de 8 bits pour représenter une valeur de satisfaction dans  $[-127, 127]$ , des bits de contrôle (début et fin de trame, parité) et un bit de détection d'obstacles. Soit un total de 16 bits par trame pour l'implémentation sur les robots :



Le schéma en figure 12.4 rappelle le principe du protocole d'allocation et de synchronisation indirecte proposé au chapitre 11. Il montre aussi la décomposition de la trame en deux parties : la première représente l'information à transmettre, la seconde, très courte, sert à la détection des obstacles. En fin de transmission d'une trame, l'agent exécute son processus comportemental (dont la durée n'excède pas celle de l'émission d'un bit).

### Principe de détection des interférences et des obstacles

- Une interférence (ou collision de forme 1, cf. 11.3.1) entre messages est détectée par l'écoute de la porteuse pendant la non émission des bits 0 de la trame, s'il y a perception de bit(s) 1 c'est qu'un autre agent est en train d'émettre,
- Une interférence à la réception (ou collision de forme 2) est détectée par vérification de la cohérence des trames reçues (début et fin de trame, parité),
- La détection d'obstacles est réalisée par l'émission de bits 1 et l'écoute simultanée de leur éventuel retour. Les signaux sont réfléchis par les obstacles, ainsi la réception d'un bit pendant l'émission est synonyme d'obstacle.

Précisons que les bits composant la trame informative ne sont pas émis avec la même intensité que ceux de la perception des obstacles. Tout d'abord, parce que l'utilisateur doit pouvoir régler la portée des communications différemment de celle de la zone de perception

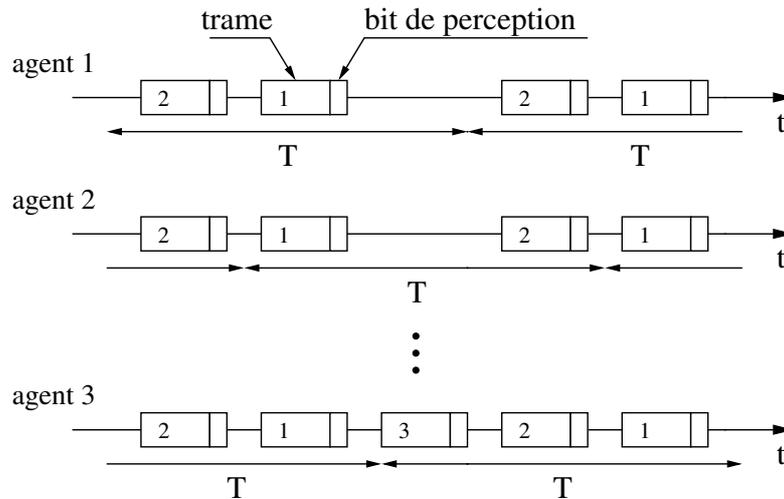


FIG. 12.4 – Principe d’allocation du protocole pour l’émission de trames et de bits pour la détection des obstacles.

des obstacles. Mais aussi parce que la détection d’obstacles nécessite un “aller-retour” des ondes entre agent et obstacle, alors que l’émission d’un message ne nécessite qu’un “aller”.

### Gestion de la période $T$ :

Le cycle de durée  $T$  assurant le fonctionnement du protocole, c’est-à-dire la synchronisation indirecte entre agents, est géré par appel direct au microprocesseur du PC. Nous avons utilisé la librairie “timer.c”, proposée par Ethan Rohrer [Rohrer, 1994], qui permet de déclarer et de lire des compteurs de temps avec une précision de 838 nano secondes. La technique employée est basée sur la lecture du décrétement d’un compteur lié au bus du PC (de fréquence 1.1931816 MHz).

La durée  $T_b$  d’émission d’un bit est de  $630 \mu s$  (1600 bauds), en deçà nous avons constaté que le bruit pouvait modifier l’information émise. La durée d’une trame  $T_e$  et du délai de séparation  $\varepsilon$  (2 bits) correspond à la durée d’émission de 18 bits, soit  $11,344 ms$ .

Pour les expérimentations menées avec 2 robots nous avons posé  $T = 3.(T_e + \varepsilon)$ , soit la durée de 54 bits, soit encore  $34 ms$ . Ce qui permet à chaque robot de communiquer des trames et de tester la présence d’obstacles à une fréquence de  $30 Hz$ .

D’autres expérimentations ont été menées avec 3 robots, nous avons pris  $T = 4.(T_e + \varepsilon)$ , soit une fréquence de communication de  $22 Hz$ .

Enfin, en supposant l’utilisation d’un système de 10 agents voisins communicants, nous obtenons une fréquence de communication de  $8 Hz$ . Cette valeur reste encore efficace, puisqu’elle assure 8 émissions et tests d’obstacles à chaque robot par seconde, et ce, quelle que soit la configuration du système. Ces différentes fréquences sont supérieures à celles

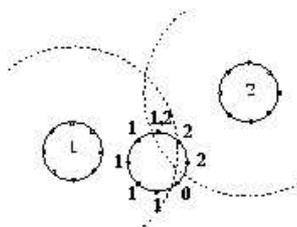


FIG. 12.5 – Réceptions indépendantes sur les huit récepteurs IR du robot

des systèmes distribués et connectés à un PC extérieur de [Goldberg and Mataric, 1999] et [Parker, 1999].

Remarque 1 : Si nous avons employé une porteuse de fréquence plus élevée, comme celles d'émetteurs HF, autorisant des taux de transfert de 4800 bauds (type Khepera), notre protocole fournirait une fréquence de communication de 24 Hz pour 10 agents (et 89 Hz pour 2 agents).

Remarque 2 : Lorsque les robots sont en action, nous avons constaté la présence d'un fort bruit dans les réceptions, dégradant les trames et provoquant de nombreuses réallocations. Pour traiter ce problème, nous avons rendu le système de réallocation moins sensible aux erreurs et instauré un système de redondance dans les transmissions.

### Lecture sur huit capteurs indépendants

Chaque robot disposant de huit récepteurs infrarouge, nous avons traité huit réceptions simultanées indépendantes. C'est-à-dire que chaque agent peut recevoir simultanément des trames différentes réparties sur les différents récepteurs (cf. schéma 12.5).

Cependant, cette possibilité est rarement utilisée puisque chaque agent est censé utiliser une période de temps différente de celle de ses voisins. Par contre, nous utilisons cette *distinction* des capteurs essentiellement pour déterminer la *direction* d'un message perçu (cf. section navigation ci-après). En terme de programmation, cela se traduit par la gestion pour chaque agent de 8 vecteurs de taille T (bits) et non d'un seul.

La programmation de l'algorithme 12.2 et des techniques de vérification des trames, ainsi que les fonctions de détection des interférences, d'allocation des zones d'émissions et de leurs glissements ne sont pas détaillées ici. Leur programmation, parfois technique, reste classique (toutefois, ces codes sont consultables sur [Simonin, 2001]).

### Distinguer les robots des obstacles

Il est essentiel dans nos applications que les agents soient capables de distinguer les robots des obstacles (non robots). Cette opération est réalisée facilement au niveau de

chaque capteur par l'exploitation des différentes perceptions (dans notre modèle un robot émet toujours une trame, car elle contient le test de détection d'obstacles) :

perception obstacle	réception trame	type d'obstacle
non	non	aucun
oui	non	obstacle (non robot)
oui	oui	robot

Cette opération de "sommation" des perceptions permet de déterminer la présence et la nature d'un obstacle proche. La *direction* d'un robot voisin est déterminée par le calcul de la moyenne des capteurs connexes actifs (précision de  $\pm 22.5^\circ$ ).

### 12.2.2 Navigation et comportement

La programmation du module de navigation a bénéficié elle aussi des expériences menées en simulations (chapitres 8 et 9). L'objectif étant d'implémenter un calcul de trajectoire capable d'appliquer l'équation vectorielle (cf. 6.2) :

$$\vec{V} = g_1 \cdot \vec{V}_{goal} + g_2 \cdot \vec{V}_{sli} + g_3 \cdot \frac{1}{M} \sum_{j=1}^M k_j \cdot \vec{V}_{alt_j}$$

Le premier terme est la direction du but recherché par l'agent, il est déduit soit de la perception d'un attracteur dans l'environnement, soit de la perception d'un signal de satisfaction interactive transformé (par altruisme) en vecteur but.

Nous détaillons ci-dessous le calcul du vecteur évitement des obstacles ( $\vec{V}_{sli}$ ). Enfin le troisième terme, de faible poids, est calculé à partir de la perception des signaux extérieurs et de la formule du vecteur altruisme.

#### Principe du calcul de trajectoire

Paramètres de contrôle :

- La vitesse instantanée  $V_i$  est calculée par la moyenne des vitesses des deux roues. Elles sont connues par la lecture des codeurs des roues,  $Vg$  et  $Vd$  sont les vitesses respectives de chaque roue,
- La direction du but courant de l'agent est exprimée par un angle  $A_{goal}$  appartenant à l'ensemble  $\{-135, -90, -45, 0, 45, 90, 135, 180\}$ , où 0 est toujours la direction de la ligne droite (cf. fig 12.6).

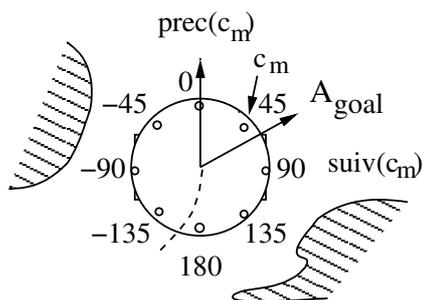


FIG. 12.6 – Schéma du robot vu de dessus,  $A_{goal}$  est la direction recherchée,  $c_m$  le capteur représentatif de cette direction.

Principe “réactif” adopté :

A chaque itération du cycle comportemental l’agent tend à se diriger vers son but, tout en évitant les obstacles et en tenant compte de ses vitesses linéaire et angulaire. L’agent évolue uniquement par application du principe répété {perceptions internes et externes, action}\*.

Algorithme général du *comportement* des robots (exécuté à chaque itération du cycle  $T$ ) :

1. évaluer les signaux perçus et une variation éventuelle de la direction du but  $A_{goal}$  (perception attracteur ou altruisme)
2. évaluer la vitesse de chaque roue par lecture des codeurs, déduire la vitesse instantanée du robot,
3. test d’arrêt d’urgence (présence d’un obstacle dans la direction de l’agent et vitesse supérieure à un seuil de non risque),
4. calculer une déviation de la direction de l’agent en fonction de la perception d’obstacles éventuels,
5. calculer et appliquer une variation de vitesse sur chaque roue en fonction des valeurs  $A_{goal}$  et  $V_i$ ,
6. (calculer la satisfaction personnelle et évaluer le signal  $I$  à émettre)

## Modules

Déviation de l’agent par perception des obstacles (étape 5) :

1. Soit  $c_m$  le numéro du capteur le plus proche de la direction  $A_{goal}$  recherchée par l’agent (cf. figure 12.6), la transformée de  $A_{goal}$  à  $c_m$  est triviale.

2. Si l'agent n'a pas déjà un sens de rotation choisir le côté de déviation (ou de glissement) le plus libre. Chaque côté est évalué par la somme des perceptions sur 3 capteurs, en pondérant les capteurs inversement à leur éloignement de  $c_m$ . Ainsi, le sens de déviation est choisi par comparaison des deux valeurs suivantes :

$$3.R(\text{prec}(c_m))+2.R(\text{prec}(\text{prec}(c_m)))+R(\text{prec}(\text{prec}(\text{prec}(c_m)))) \text{ (gauche)}$$

$$3.R(\text{suiv}(c_m))+2.R(\text{suiv}(\text{suiv}(c_m)))+R(\text{suiv}(\text{suiv}(\text{suiv}(c_m)))) \text{ (droite)},$$

où  $\text{prec}(c)$  renvoie le numéro du capteur précédent le capteur  $c$ ,  $\text{suiv}(c)$  le suivant et  $R(nc)$  l'information sur le capteur numéro  $nc$  (0 ou 1).

3. Si  $R(c_m)=1$  Alors calculer une variation de direction de  $30^\circ$  dans le sens défini précédemment

4. ajouter des variations suivants les perception des capteurs adjacents à  $c_m$  :

si $R(\text{prec}(c_m))=1$	$+12^\circ$
si $R(\text{prec}(\text{prec}(c_m)))=1$	$+10^\circ$
si $R(\text{suiv}(c_m))=1$	$-12^\circ$
si $(\text{suiv}(\text{suiv}(c_m)))=1$	$-10^\circ$

Ces différentes valeurs ont été réglées expérimentalement. Cependant, la fréquence du cycle perception-action est suffisamment élevée pour transformer ce contrôle du discontinu en un comportement continu plus proche d'un asservissement proportionnel.

Commande sur les roues (étape 6) :

– Le produit  $\rho$  de la vitesse instantanée par l'angle recherché (soit  $V_i \cdot |A_{goal}|$ ) est utilisé pour déterminer les types d'actions :

arrêt d'urgence ou $\rho > \rho_{max}$	stopper : $V_g = 0$ et $V_d = 0$
si $A_{goal} = 0$	incrémenter $V_d$ et $V_g = V_d$
si ( $\rho < \rho_{max}$ et $ A_{goal}  \leq 30^\circ$ )	si $A_{goal} > 0$ accélérer $V_g$ et diminuer $V_d$ prop. à $ A_{goal} $
	si $A_{goal} < 0$ accélérer $V_d$ et diminuer $V_g$ prop. à $ A_{goal} $
Sinon	rotation : $V_g = -V_d$

Certaines situations exigent un arrêt immédiat (perception d'un obstacle à vitesse élevée, angle recherché trop important en rapport à la vitesse). Ce qui est intéressant dans un tel modèle à règles réactives, c'est que l'enchaînement d'une rotation (dernière ligne) après le déclenchement d'un arrêt d'urgence est fluide et sans "cassure" (la rotation commence alors que l'agent n'a pas terminé de s'arrêter).

Enfin les virages pris par le robot se font par accélération de la roue extérieure au virage

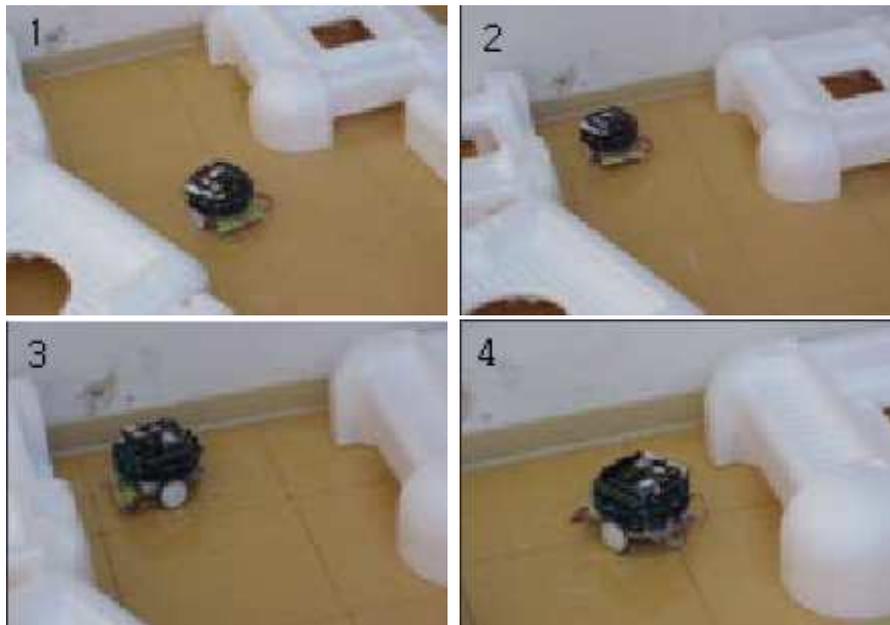


FIG. 12.7 – Exemple d'évolution d'un robot dans un environnement contraint

et par décélération de la roue intérieure au virage (proportionnellement à  $V_i$  et  $|A_{goal}|$ ).

Le résultat en expérimentation montre à la fois les capacités réactive (rapidité des réaction) et de composition (des perceptions) du modèle. Le déplacement du robot est fluide et adapté à son environnement et ses éventuels changements (obstacle humain, autres robots, etc.).

La figure 12.7 donne un exemple d'un robot évoluant dans un espace réduit, l'agent s'arrête prêt du mur tout en enchaînant une rotation, suivie d'une progression lente devant les obstacles, jusqu'à pouvoir trouver un espace libre et accélérer.

## 12.3 Expérimentation du modèle satisfaction-altruisme

### 12.3.1 Programmation du modèle Satisfaction-Altruisme sur les robots

Nous avons présenté en première partie de ce dernier chapitre les modules de communications et de navigation des robots. Nous précisons maintenant l'écriture du modèle de comportement altruiste et montrons son efficacité sur le problème des conflits dans les environnements en impasses.

Les robots évalués ont pour seule tâche de se déplacer aléatoirement et d'insister dans une direction lorsqu'ils sont face à un individu de leur "espèce". Le module de prise de décision implémenté correspond à l'algorithme donné tout au long de la thèse (section 7.2.1).

### Mesure de la satisfaction

Nous avons implémenté la mesure proposée au chapitre 9 section 9.5.1 (résolution de conflits). Il s'agit d'évaluer pour chacun des capteurs le type d'obstacle présent, puis de pondérer l'insatisfaction suivant ce critère.

Si la vitesse de déplacement de l'agent est supérieure à la vitesse minimale de "progression normale" alors  $P \leftarrow P + 0.2$

Sinon Pour chaque capteur

si (obstacle = mur) alors  $P \leftarrow P - 0.03$

sinon si (obstacle = robot) alors  $P \leftarrow P - 0.01$

sinon  $P \leftarrow P + 0.05$  (pas d'obstacle)

La valeur de satisfaction évolue dans l'intervalle de valeurs entières [-127, 127]. Lorsqu'un agent est insatisfait ( $P < 0$ ), il émet un signal répulsif  $I$  de valeur égale à  $P$ . Le coefficient d'altruisme vaut 0.5. Les quatre règles définissant le modèle avec capacité à résoudre les conflits d'encombrements, données en section 9.4.2, ont été impémentées telles quelles.

### 12.3.2 Résolution du problème du corridor fermé

Nous avons débuté les expériences de résolution de conflit avec seulement deux robots. La raison essentielle à cela était de parvenir à mettre au point l'ensemble des modules (communication, perceptions, navigation, évaluation de la satisfaction) sans se perdre dans la complexité des paramètres et des pannes/erreurs/bruits croissant "exponentiellement" en fonction du nombre de robots.

Malgré cela nous avons rencontré de nombreux problèmes, inhérents à tout système réel, mais déconcertants pour un informaticien.

Nous avons évoqué plus haut les problèmes liés aux communications, que nous avons en partie résolus. L'autre point délicat se situe au niveau de la perception des obstacles par les robots. Les difficultés proviennent de la diffusion peu régulière des signaux infrarouges (en

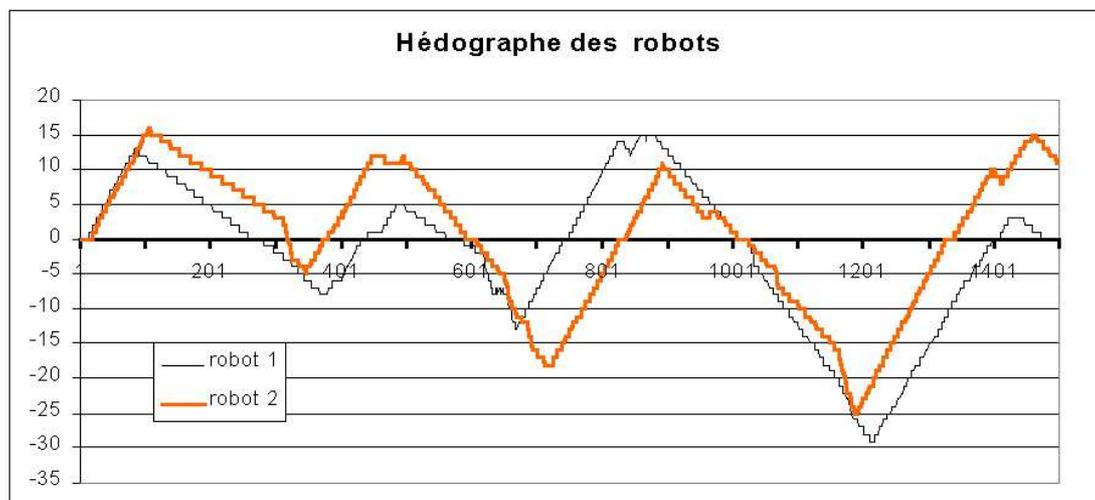


FIG. 12.8 –

distance) et aux différentes réflexions liées à la nature des matériaux détectés et surtout à leur couleur. Pour limiter ces problèmes, nous avons construit un couloir à partir de chutes d’emballages en polystyrène de couleur blanche, qui ont aussi l’avantage de ne pas endommager les robots en cas de chocs intempestifs.

Nous avons volontairement “fermé” les deux extrémités du couloir pour créer deux impasses. Le but étant de ne pas limiter l’expérience à une seule résolution (sortie), mais bien de voir se reproduire le phénomène plusieurs fois. Plus exactement, d’observer une oscillation des deux robots se repoussant à tour de rôle, d’une extrémité à l’autre du couloir.

### Résultats

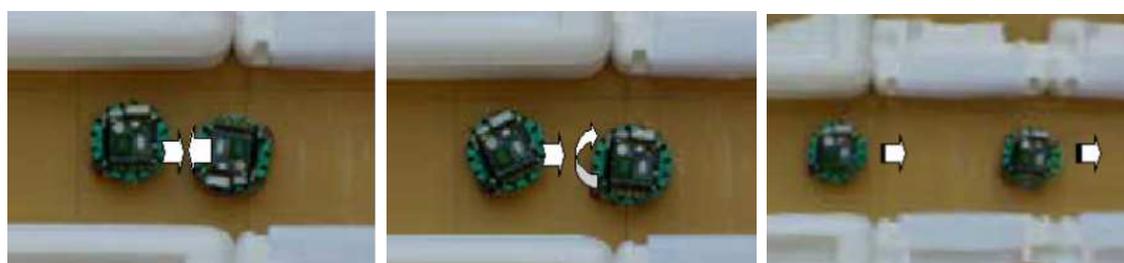
De nombreuses expériences ont été filmées, nous en rapportons une en détail décrivant le comportement souvent observé des deux robots (cf. [Simonin, 2001]).

Pour mieux analyser la résolution et pouvoir la présenter, nous avons enregistré dans chaque robot l’évolution des principaux états (satisfaction personnelle, état d’altruisme, signaux émis et captés).

Les photos (du film) présentées ci-après décrivent l’évolution de l’expérience, elles sont analysées en s’appuyant sur le tracé des hédographes superposés des deux agents figure 12.8 (qui donne aussi les valeurs de signaux émis puisque pour  $P < 0$ ,  $I = P$ ). Celui-ci montre clairement l’alternance des niveaux d’insatisfactions prioritaires atteints par les agents au cours de l’expérience (courbes proches de celles obtenues en simulations, cf. section 9.5.2 fig. 9.6).



1. Les deux robots, positionnés initialement aux extrémités du couloir avancent jusqu'à se rencontrer (leur sat.  $P \nearrow$ ), période  $[0,100]$  figure 12.8. Lorsqu'ils insistent pour continuer devant eux, leur satisfaction respective chute, période  $[100,300]$ .



2. Dès que leur insatisfaction chute sous 0, les deux robots émettent un signal répulsif. Le robot 1 (à gauche) étant plus insatisfait (période  $[250,400]$ ) il parvient à faire passer le robot 2 (à droite) en état d'altruisme. Alors celui-ci réalise une rotation (les satisfactions chutent toujours), puis il s'éloigne, suivi du robot 1 (les satisfactions remontent).



3. Les deux robots arrivent à l'extrémité droite du corridor. Celui qui découvre sa fermeture, tourne sur lui même pour tenter de repartir, mais il se trouve alors face au robot 1 (leur satisfaction chute, période  $[500,600]$ ). Les insatisfactions sont proches, mais le robot 2 plus entouré de murs que le premier, parvient à déclencher une réaction altruiste chez son voisin. Le robot 1 change de but, il tourne sur lui même et part dans une nouvelle direction  $[600,700]$ . A ce moment là, la satisfaction des agents cesse de chuter pour se remettre à monter  $[700,800]$ .

Le même processus de conflit, puis de résolution, se produit à l'autre extrémité du couloir (cf. fig. 12.8 période [1000,1200]). Les deux robots oscillent ainsi d'un bout à l'autre du couloir plusieurs fois.

L'expérience réalise bien les comportements attendus, et ce malgré de nombreuses coupures dans les communications (jusqu'à 40% de pertes sur certaines périodes). Le système résiste bien car le niveau de satisfaction calculé par les agents ne dépend que de la perception des types d'obstacles. De plus, il suffit de peu de trames reçues pour qu'un agent réalise une rotation (persistance de courte durée) et donc change de direction-but.

### Propagation des répulsions (3 robots)

Nous avons débuté de nouvelles expériences avec un troisième robot, le but étant de vérifier la propagation des répulsions entre agents. Ceux ci ont été placés dans le couloir décrit précédemment.

Nous avons observé le processus "de réaction en chaîne", où le robot le plus près d'une extrémité repousse le second qui lui même repousse le troisième. Mais ce couloir est trop court pour réaliser correctement l'expérience, car les deux robots placés aux extrémités de la chaîne se trouvent dans la même situation. Nous avons aussi observé une certaine instabilité dans ce processus, due essentiellement aux erreurs de transmissions (encore trop nombreuses) et à l'insuffisance des écarts entre les (in)satisfactions personnelles (lié aux perceptions délicates à base d'infrarouges). Nous travaillons actuellement à régler ces paramètres, afin d'obtenir un comportement plus robuste.

## 12.4 Bilan

Ces premiers résultats expérimentaux nous apparaissent très satisfaisants car ils valident plusieurs de nos objectifs.

- Le module de navigation réactive, intégré à la boucle de communication et de détection des obstacles, fonctionne efficacement. Les robots sont capables d'évoluer *rapidement* dans un environnement dynamique (parmi d'autres robots) et sans chocs.

- Le principe du protocole de communication proposé a été validé expérimentalement. Il permet à des robots totalement autonomes d'échanger des informations par diffusion

locale de courtes trames à des fréquences de 8Hz à 30Hz. Nous l'avons en particulier testé dans la situation la plus extrême, celle d'émissions incessantes par tous les robots. Dans ces conditions nous avons observé un taux d'erreur de transmission très variable (de 0 à 40%), certainement du à la nature du médium utilisé, c'est-à-dire aux signaux infrarouges.

En contrepartie le protocole proposé nous a permis d'intégrer les communications et la détection d'obstacles (dont leur nature) par un unique système d'émission/réception peu onéreux.

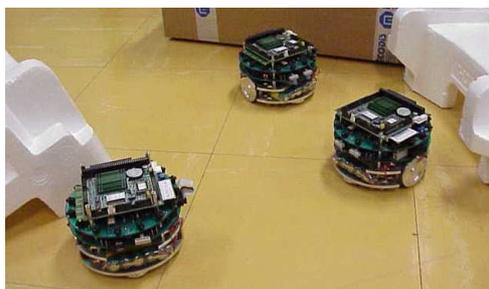
- Le modèle satisfaction-altruisme, tant au niveau de l'évaluation de la satisfaction de l'agent que du principe de résolution des conflits a été validé expérimentalement sur deux robots. Nous avons bien retrouvé des comportements analogues à ceux obtenus en simulation (les hédographes montrent une similitude des variations de niveaux de satisfactions en fonction des perceptions locales). Tout agent bloqué dans une impasse devient plus insatisfait que son voisin et parvient à le repousser, c'est-à-dire à le faire passer en état d'altruisme.

Les expériences avec trois robots montrent que le principe de propagation des insatisfactions fonctionne, mais les modules de communications et de perceptions doivent être affinés pour parvenir à une résolution complète.

Toutefois, malgré des "coupures" dans les communications, la résolution à deux robots est apparue robuste. Ceci nous laisse penser que nous pourrions généraliser ce résultat à plusieurs unités.

Plus généralement, lors de nos nombreuses expériences, nous avons souvent été étonné d'observer des comportements de robots s'apparentent à des attitudes du "vivant" !

En effet, lorsque les robots se rencontrent, dans un premier temps ils se reconnaissent (par échanges de signaux), puis s'ils désirent utiliser le même lieu de passage une attitude de compétition s'instaure. Les robots réalisent quelques déplacements mais restent face à face, c'est la phase d'impatience et d'insistance pour passer. Dès que l'un d'eux parvient à exprimer une insatisfaction suffisante, l'autre change d'attitude et coopère pour le bien des deux. Les robots sortent du conflit, puis retrouvent un comportement individuel normal.





## Sixième partie

### Conclusion et Annexes



# Chapitre 13

## Conclusion

### 13.1 Bilan

Nous avons défini au premier chapitre notre problématique générale sous la forme d'une question : comment introduire une forme de coopération intentionnelle dans l'approche réactive ?

La thèse a été construite comme le développement d'une réponse. Nous reprenons les grandes étapes qui la composent et nous précisons nos différentes contributions.

Dans un premier temps, nous avons mené une étude approfondie de la notion de coopération entre agents situés. Celle-ci nous a conduit à dégager quatre grandes catégories de situation/action coopérative : aider les actions d'autrui, partager les ressources et éviter / résoudre les conflits. Nous avons alors précisé les trois capacités nécessaires aux agents pour qu'ils puissent les réaliser : (i) être capable de percevoir une telle situation (ii) pouvoir communiquer/influencer ses voisins, (iii) être capable d'agir coopérativement.

Pour intégrer ces trois capacités dans un modèle réactif, nous avons tout d'abord reconsidéré la notion de satisfaction chez l'agent situé. Ainsi, nous avons défini un niveau de *satisfaction personnelle*, basé sur une mesure incrémentale de la progression/stagnation des actions de l'agent (extension des "progress estimators" de [Mataric, 1994a]). Cette modélisation donne la possibilité à un agent d'évaluer sa satisfaction suivant l'exécution de ses actions *individuelles*. Pour compléter cette évaluation, nous avons introduit la notion de *satisfaction interactive*, dédiée uniquement à la perception des actions des agents voisins (aide, gêne, indifférence).

A partir de ces deux aspects individuel et relationnel, nous avons ré-exprimé les situations nécessitant des actions coopératives en termes de satisfactions et d'insatisfactions personnelles ou interactives. Mais aussi en définissant les *réactions* adéquates à ces situations par des *actions spatiales, pour conserver une approche réactive*.

Pour établir la liaison entre perception d'un besoin par un agent et réaction/coopération d'un autre, nous avons introduit un moyen de *communication minimal et local*, basé sur la diffusion de *signaux répulsifs ou attractifs*. Ces signaux codent des valeurs *homogènes avec le niveau de satisfaction personnelle* des agents.

Par conséquent, nous avons pu poser une condition de réaction aux signaux perçus, nommée *test d'altruisme*. Cette condition permet de garantir un comportement coopératif de la part des agents les moins contraints vis à vis des autres. Ce résultat est obtenu en définissant le comportement des agents comme une compétition entre la poursuite de la tâche courante et la réaction aux influences extérieures. Le test d'altruisme, comparant la satisfaction personnelle aux signaux perçus, décide de ce choix.

Dans le but de conserver un modèle d'agent réactif, la réaction altruiste est définie comme la transformation immédiate de la valeur perçue en un *vecteur d'altruisme* (attractif ou répulsif). Ainsi, l'agent évolue en combinant un vecteur but (le vecteur d'altruisme lorsqu'il est dans cet état), l'évitement des obstacles et l'évitement des agents émettant des signaux répulsifs.

L'apport essentiel de ce travail est donc l'introduction de comportements coopératifs intentionnels et de réactions altruistes à l'approche réactive, sans que les principes d'auto-organisation et de robustesse soient remis en cause. L'intégration de l'ensemble de ces capacités permet de développer des systèmes multi-agents situés coopératifs capables de traiter efficacement des problèmes physiquement distribués, et en particulier, d'assurer la résolution de problèmes de navigation. Ceci n'a pu être possible qu'en définissant des influences sous formes de signaux locaux, interprétés en temps réel par le module de sélection/combinaison d'actions des agents. Mais aussi parce que ces signaux ont une expression homogène avec la mesure de satisfaction personnelle des agents.

Cette proposition d'architecture, à base de comportements simples déclenchés par des perceptions de l'environnement et des signaux émis par les agents, nous a conduit à mener les travaux suivants afin d'en analyser les performances :

- L'application du modèle au problème des robots fourrageurs s'est avérée simple et efficace. L'étude menée par simulation a montré que les signaux échangés amplifiaient certains comportements d'auto-organisation et par conséquent réduisaient les temps de

résolution. La spécification de relations simples entre comportements et signaux nous a permis d'établir une solution robuste et adaptative, dont les performances sont proches des temps optimaux.

- En précisant la définition des signaux répulsifs émis par les agents, nous avons pu montrer l'efficacité du modèle à résoudre des conflits spatiaux. Cette étude nous a permis de montrer que le noyau de propagation des répulsions était équivalent à un processus d'éco-résolution. Mais pour parvenir à une solution applicable à des agents situés, nous avons introduit les notions de *persistance* des répulsions et *d'amplification* des insatisfactions, présentes dans le modèle satisfaction-altruisme. Il s'agit donc d'une extension du principe d'éco-résolution par l'introduction de ces notions, d'une mesure continue de la satisfaction et de l'émission/réaction de signaux attractifs.

- La troisième application a permis d'évaluer le modèle sur un système d'agents hétérogène. Cette étude montre la généralité du modèle mais aussi certaines de ses limites, comme l'incapacité des agents à retenir/reproduire les situations/actions coopérativement satisfaisantes. Pour pallier ce défaut, nous avons intégré à l'architecture un module d'apprentissage permettant de renforcer positivement ou négativement les poids de déclenchement des comportements.

Pour cela, nous avons exploité le modèle des satisfactions pour (i) modéliser l'espace des situations d'interactions par une représentation très compacte et (ii) calculer la récompense comme une fonction des valeurs des signaux perçus. Les résultats montrent l'apparition de nouvelles fonctionnalités chez les agents, mais surtout la diminution de celles qui, pour certaines interactions, sont néfastes.

L'étude des applications présentées ci-dessus nous a amené à développer des outils de simulation permettant d'observer et d'analyser l'activité des systèmes (en terme de satisfactions/insatisfactions et d'influences entre agents). Ces réalisations sont autant d'outils génériques que nous comptons utiliser pour l'étude d'autres SMA situés.

### Validations expérimentales

L'un des objectifs de ce travail était d'élaborer un modèle d'agents situés (coopératifs) qui soit réellement exécutable sur des robots autonomes. En effet, les contraintes présentes dans les systèmes réels ne sont pas toujours prises en compte par les travaux menés en informatique (pannes, erreurs, bruit, dépendances aux conditions environnementales).

Notre démarche, consistant à intégrer ces contraintes dès le début de notre étude, nous a conduit à l'élaboration d'un modèle d'agents coopératifs robuste et implémentable.

Ces réalisations nous ont permis d'atteindre un objectif relativement ambitieux, celui de valider notre modèle et les hypothèses posées par l'expérimentation sur des robots réels. Et plus particulièrement, de valider le principe de résolution des conflits spatiaux.

L'une des conditions fortes posées à l'application du modèle est la nécessité d'assurer aux agents des échanges incessants de signaux à portée locale. Pour répondre à ce problème, nous avons, là aussi, développé un protocole de communication dédié aux systèmes multi-agents situés.

Notre approche, intégrant d'emblée les notions de localité et de mobilité, nous a permis de proposer une solution valide pour des systèmes d'agents *mobiles autonomes ouverts*. En effet, en établissant une forme de synchronisation indirecte entre les agents, nous avons pu développer un système de réservation de zones temporelles d'émissions sans collisions (type multiplexage temporel). Le résultat présente l'avantage d'assurer un taux de communication constant, qui ne dépend que du voisinage maximum de communication d'un agent, donc qui est indépendant du nombre total d'agents.

Ce protocole a été implémenté et validé lors de nos expérimentations avec les mini-robots mobiles proposés. Nous avons montré aussi que la détection d'obstacles pouvait être intégrée directement au système de communication. Ceci a été réalisé et validé par l'emploi de seulement 16 émetteurs et 8 récepteurs infrarouges.

## 13.2 Perspectives

Nous nous sommes concentré, au cours de cette thèse, sur des problèmes concernant des systèmes multi-agents *situés*. Cependant, les notions de satisfaction personnelle et interactive peuvent être étendues à d'autres types d'agents ou SMA.

En particulier, il serait intéressant de rapprocher/analyser les notions de satisfactions en termes d'états psychologiques d'un agent cognitif. Ceci dans le but d'établir un modèle d'émotion et/ou d'apprentissage lié à la progression des actions menées par l'agent.

Mais le champ d'application le plus immédiat nous semble être la définition ou l'analyse de sociétés d'agents. En effet, si chaque agent évalue une satisfaction personnelle et interactive, il est possible de percevoir/contrôler le système en terme de satisfactions/insatisfactions. Nous avons débuté un travail de formalisation des situations coopérantes, compatibles, incompatibles et conflictuelles à base seulement de niveaux de satisfactions. Nous espérons obtenir à un modèle générique, qui soit applicable à divers types de SMA.

L'étude de l'application des processus d'éco-résolution aux agents situés doit être poursuivie et mieux formalisée. Notre objectif étant d'aboutir à une véritable définition et implémentation d'éco-robots.

En contrepartie, nous pensons que l'intégration de signaux attractifs entre éco-agents peut être pertinente pour la résolution de problèmes. Pour l'instant, nous n'en avons qu'une illustration dans les processus de recrutements entre agents situés (altruistes). Nous pensons poursuivre ces réflexions par une étude plus formelle de cette proposition.

Il apparaît à travers les problèmes traités (robots fourrageurs, résolution de conflits spatiaux) que la notion de persistance soit nécessaire à l'accomplissement de certaines tâches. Cette notion est présente à la fois dans la mesure de la satisfaction personnelle (notion de persévérance en cas d'insatisfaction) et dans l'émission des signaux répulsifs et attractifs (règles de résolution des conflits et comportement d'attente des robots fourrageurs). Nous pensons qu'il s'agit d'un concept manquant au modèle réactif pour assurer la réalisation correcte de certains comportements individuels ou collectifs. Cette réflexion, tout juste engagée, devrait être au centre de nos prochains travaux concernant les modèles réactifs.

L'élobaration d'un modèle *d'agents apprenants* basé sur des échanges de satisfactions se poursuit. Les premiers résultats nous apparaissent prometteurs et devraient être étendus dans les prochains mois.

Enfin, dans une perspective de développement de "cyber-robots", nous comptons utiliser les principes de notre architecture pour modéliser des comportements sensori-moteurs humains.

Sur le plan expérimental, nous allons continuer nos travaux avec des populations de robots plus importantes. Dans ce cadre, nous traiterons également des problèmes mettant en jeu des ensembles hétérogènes de robots.



# Chapitre 14

## Annexes

### 14.1 Simulations

#### 14.1.1 Robots fourrageurs

##### Capacités des robots

vitesse de déplacement constante : largeur d'un pixel (u) / itération  
durée totale du chargement de la benne : 30 itérations,  
autonomie des batteries : 1500 u = déplacements durant 1500 itérations,  
seuil de déclenchement du retour à la base : 500 u restantes,  
débit de recharge des batteries à la base : 10 u par itération,

##### La marche aléatoire

A chaque itération (ou pas des robots), le déplacement aléatoire est réalisé par un éventuel changement de direction suivant une probabilité de 0.5 :

if  $\text{random}(1) > 0.5$  alors calculer une variation de direction *aléatoire* comprise dans  $[-22.5^\circ, +22.5^\circ]$ .

#### 14.1.2 Calcul de trajectoire des éco-robots

Chaque agent perçoit localement son environnement sur le secteur  $[-90^\circ, 90^\circ]$  devant lui sur une portée maximale de  $r_p = 30$  cm ( 30 cm = longueur d'une ligne horizontale de 30 pixels).

Principe : l'agent somme la perception des divers obstacles pour en déduire un vecteur répulsion. Celui-ci est transformé en une force de glissement qui est appliquée comme une rotation du vecteur direction courant.

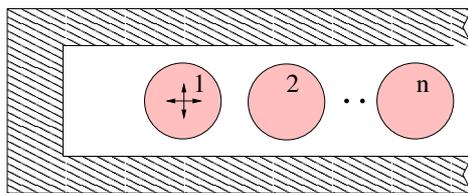
Application formule 6.2 :

1. calcul du vecteur but  $\vec{V}_g(N, \Theta)$  (vecteur agent-but)
2. intégrer l'influence d'éventuels signaux répulsifs (coefficient  $g_3$  tel que la variation d'angle de  $\vec{V}_g$  soit au plus de  $\pm 18^\circ$ )
3. perception des divers obstacles : moyenne des vecteurs (agent-obstacle) donne  $\vec{V}_r(N', \Theta')$ , il est transformé en répulsion :  $\vec{V}_r \leftarrow (N_{min}, \Theta' + 180^\circ)$  de norme égale à la distance du plus proche obstacle  $N_{min}$ ,
3. application de  $\vec{V}_r$  sur  $\vec{V}_g$  comme une force de glissement :  

$$\vec{V}_g \leftarrow (N, \Theta + S(\vec{V}_r, \vec{V}_g) \cdot f(\|\vec{V}_r\|))$$
 avec  $S$  une fonction déterminant le sens de rotation de  $\vec{V}_g$ , suivant le signe de la différence des angles de  $\vec{V}_g$  et  $\vec{V}_r$ ,  
 $f$  est une fonction qui transforme la norme de  $\vec{V}_r$  (donc la proximité à l'obstacle) en une intensité de rotation variant de façon inversement proportionnelle à sa valeur (entre  $0^\circ$  et  $180^\circ$ ).

## 14.2 Résolution des conflits spatiaux

### 14.2.1 Preuve de résolution pour le problème de l'impasse



Soit un ensemble d'agents  $a_1, \dots, a_n$  situés initialement à l'extérieur du couloir représenté ci-dessus. L'énoncé du problème est donné en section 9.5.1.

Supposons qu'à  $t = 0$  un premier agent entre dans le corridor (noté  $a_1$ ). D'autres agents peuvent au cours du temps entrer dans le corridor :  $a_2, \dots, a_n$ .

Tant que  $a_1$  progresse vers le fond du corridor, tous les agents dans le couloir en font de même et leur satisfaction personnelle sont élevées ( $\forall i P_{a_i}(t) > 0$ ).

Lorsque  $a_1$  découvre le fond du corridor, son but change, il cherche à en sortir. Rapidement  $a_1$  et  $a_2$  se rencontrent et se bloquent (conflit d'accès), leur satisfaction personnelle chute. D'autres agents explorant le corridor peuvent "buter" sur  $a_2$  (instant noté  $t_2$ ). Ces agents  $a_3, \dots, a_{k \leq n}$  bloqués ont leur satisfaction personnelle qui chute aussi. Ils sont dans la même situation que  $a_2$ , mais leur immobilisation arrivant nécessairement plus tard assure que  $\forall t > t_0 \forall j > 2, P_{a_2}(t) < P_{a_j}(t)$  (notée inégalité 1).

Lorsque la satisfaction personnelle de  $a_1$  ou  $a_2$  devient négative, l'agent émet un signal répulsif égal à  $P$ , deux situations sont possibles :

1.  $a_1$  est le premier à émettre :  $a_2$  perçoit la répulsion  $I_{1/2}$ , plus forte que la sienne, il passe en état d'altruisme.  $a_2$  tente alors de sortir du corridor. Les agents  $a_3, \dots, a_k$  n'étant pas plus insatisfaits que  $a_2$  (inégalité 1), nous avons  $\forall j > 1, I_{1/2} < P_{a_j}(t)$ . Donc la répulsion de  $a_1$  se propage jusqu'à  $a_k$ . Ces  $k$  agents prennent la direction de la sortie.

Si le  $k$ ème agent rencontre un agent  $a_{k+1}$  explorant le corridor ( $P_{a_{k+1}}(t) > 0$ ), l'insatisfaction propagée par  $a_k$  fait immédiatement passer  $a_{k+1}$  en altruisme, augmentant la chaîne d'agents. Ce processus se répète jusqu'à ce que les agents  $a_2, \dots, a_n$  sortent du corridor, libérant  $a_1$ .

2.  $a_2$  est le premier à émettre :  $a_1$  perçoit la répulsion  $I_{2/1}$  et se trouve repoussé par altruisme vers le fond du corridor. Bloqué, son insatisfaction augmente fortement jusqu'à dépasser l'intensité du signal  $I_{2/1}$ . Ainsi  $a_1$  peut sortir de son état d'altruisme et peut émettre un signal répulsif nécessairement plus "fort" que celui de  $a_2$ . L'inégalité 1 assure alors que  $\forall j > 1, I_{1/2} < P_{a_j}(t)$ . Nous retrouvons une situation identique à celle du cas précédent (1), qui garantit la résolution du problème.

Pour éviter que deux agents voisins ne se retrouvent avec des valeurs égales à  $-P_{max}$ , nous avons rajouté à ces derniers de petites variations aléatoires dans leur évaluation de la satisfaction personnelle. On est ainsi assuré que le système se trouve toujours dans l'un des deux cas précédents.

Si le nombre d'agents dans le couloir est important, il se peut que  $a_1$  ne puisse repousser tous les autres agents à cause du processus d'amortissement des signaux. Or, dans ce cas, son insatisfaction continue à augmenter et plus vite que les autres. Ainsi, la force du signal répulsif propagé s'amplifie. Cette amplification se poursuit jusqu'à ce que tous les agents puissent être repoussés, c'est-à-dire quand  $\forall j > 1, P_{a_1} < P_{a_j}(t)$ .

Ce même processus garantit la résolution dans le cas où la chaîne d'agents rencontrerait un agent isolé  $a_e$  très insatisfait. Si  $a_1$  doit retourner au fond du couloir, l'amplification de son insatisfaction permettrait de repousser tous les agents dont  $a_e$ .

### 14.3 Simulation du protocole de communication

Nous nous sommes focalisé sur des situations de fortes charges du réseau, c'est-à-dire lorsque tous les agents présents sont proches et tentent de communiquer simultanément (c'est le cas dans nos expérimentations en robotique). Nous n'avons pas simulé le processus de brouillage car les agents sont considérés suffisamment proches pour communiquer tous ensembles directement.

L'application prend en entrée le nombre d'agents que l'on souhaite faire interagir, la durée d'une trame  $T_e$ , la durée de la période  $T$  et le paramètre  $p$ . Le simulateur affiche alors un ensemble de lignes horizontales, une par agent, de longueur  $2.T$  (fig. 14.1.a). Pour représenter l'asynchronisme des agents, les périodes  $T$  sont déphasées d'une durée prise aléatoirement. La période de la première ligne débute toujours à  $t=0$ , les autres ont une origine différente et quelconque.

L'algorithme général exécuté par le simulateur est le suivant :

---

Pour chaque agent faire un tirage aléatoire d'une zone d'émission.

Pour chaque pas de temps  $t$  faire

  Pour chaque agent  $a_i$  faire

    Pour chaque agent  $a_j$  voisin de  $a_i$  faire

      | Si  $a_j$  émet à  $t$  Alors marquer sur la période de  $a_i$  que le pas de temps est occupé

      | Si  $a_i$  et  $a_j$  émettent à  $t$  Alors collision :  $a_i$  exécute le module de réallocation

      | Sinon exécuter le module de glissement

---

Dans un premier temps, l'algorithme alloue aléatoirement à chaque agent une zone d'émission (représentée en noir), ceci dans le but de provoquer plusieurs collisions. La figure 14.1.a montre l'état de la simulation à ce stade, les zones libres sont en blancs. Il est ainsi très facile de visualiser les collisions entre agents, ce sont plusieurs zones d'émissions qui se trouvent sur une même ligne de temps.

Pour représenter l'évolution du temps, nous avons affiché une ligne verticale, nommée *ligne de temps*, qui progresse régulièrement tout au long de la simulation (cf. 14.1.b). Comme les agents gèrent une boucle temporelle, la ligne de temps traverse de gauche droite la fenêtre et réitère cette opération jusqu'à l'arrêt du programme.

A mesure qu'une période est balayée, l'agent stocke les zones occupées, qui sont affichées en gris sur les snapshots. Lorsqu'un agent sort de sa zone d'émission, il efface toutes ses

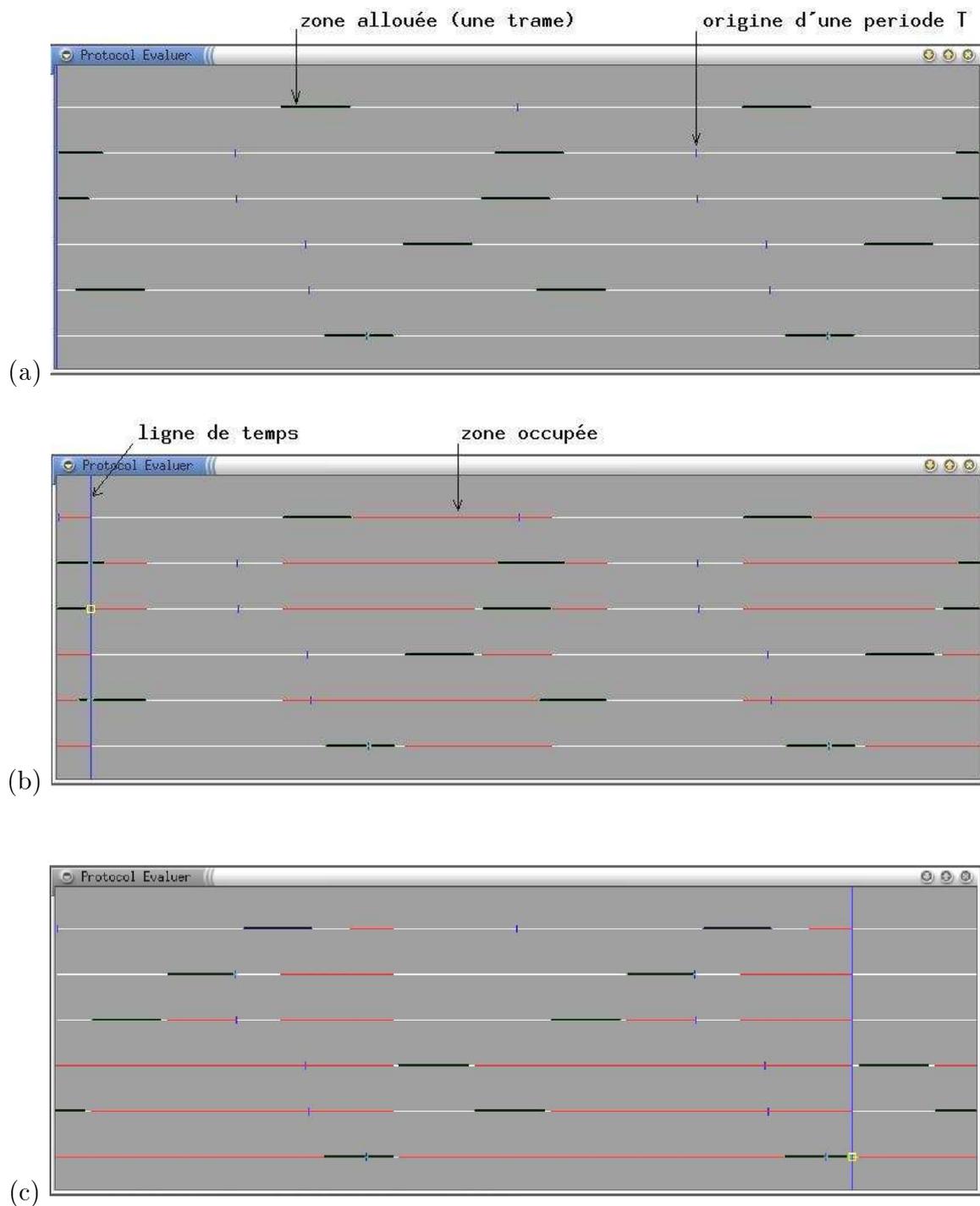


FIG. 14.1 – Snapshots (a) état initial d'une simulation avec 6 agents (b) après 2 itérations, chaque agent a détecté les zones occupées (c) après 8 itérations chaque agent a trouvé une zone d'émission sans collisions.

informations de zones occupées (remise en blanc des lignes).

La figure 14.1.b montre l'état du système après deux itérations. Les itérations suivantes provoquent des réallocations et des glissements. A la huitième itération le système a convergé, chaque agent peut émettre des trames à ses voisins sans collisions (figure 14.1.c).

# Bibliographie

- [Alami et al., 1998] Alami, R., Fleury, S., Herrb, M., Ingrand, F., and Robert, F. (1998). Multi-robot cooperation in the martha project. *IEEE Robotics ans Automation Magazine*, 5(1) :36–47.
- [Arkin, 1987] Arkin, R. (1987). Motor schema based navigation for a mobile robot : An approach to programming by behavior. *Proceedings of the IEEE Conference on Robotics and Automation*, pages 264–71.
- [Arkin, 1989] Arkin, R. (1989). Motor schema based mobile robot navigation. *International Journal of Robotics Research*, pages 92–112.
- [Arkin, 1992a] Arkin, R. (1992a). Cooperation without communication : Multiagent schema-based robot navigation. *International Journal of Robotic Systems*, 9(3) :351–364.
- [Arkin, 1992b] Arkin, R. (1992b). Homeostatic control for a mobile robot : Dynamic replanning in hazardous environnements. *International Journal of Robotic Systems*, 9(2) :197–214.
- [Arkin, 1998] Arkin, R. (1998). *Behavior Based Robotics*. The MIT Press.
- [Arkin and Balch, 1997] Arkin, R. and Balch, T. (1997). Aura : Principles and practice in review. *Journal of Experimental and Theoretical Artificial Intelligence*, 9 :175–189.
- [Arkin and Balch, 1998] Arkin, R. and Balch, T. (1998). *Cooperative Multiagent Robotic Systems, chapter in Artificial Intelligence and Mobile Robots*. D. Kortenkamp, R.P. Bonasso and R. Murphy (eds), MIT Press.
- [Asada et al., 1998] Asada, M., Stone, P., Kitano, H., Werger, B., Kuniyoshi, Y., Drogoul, A., Duhaut, D., Veloso, M., Asama, H., and Suzuki, S. (1998). The robocup physical agent challenge : Phase I. *Applied Artificial Intelligence*, 12, 2-3 :251–264.
- [Balch and Arkin, 1994] Balch, T. and Arkin, R. (1994). Communication in reactive multi-agent robotic systems. *Autonomous Robots*, 1 :27–52.

- [Balch and Arkin, 1995] Balch, T. and Arkin, R. (1995). Motor schema-based formation control for multiagent robot teams. In *Proceedings 1995 International Conference on Multiagent Systems*, pages 10–16, San Fransisco, CA.
- [Beni and Wang, 1989] Beni, G. and Wang, J. (1989). Swarm intelligence in cellular robotic systems. In *Proceedings of the NATO Advanced Workshop on Robots and Biological Systems*, Il Ciocco, Tuscany, Italy.
- [Bonabeau and Theraulaz, 1994] Bonabeau, E. and Theraulaz, G. (1994). *Intelligence Collective*. Editions Hermes.
- [Bouron, 1992] Bouron, T. (1992). *Structures de communication et d'organisation pour la coopération dans un univers multi-agents*. Thèse de l'université Paris 6.
- [Braitenberg, 1984] Braitenberg, V. (1984). *Vehicles : Experiments in Synthetic Psychology*. MIT Press , Cambridge, MA.
- [Brooks, 1986] Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1) :16–23.
- [Brooks, 1990] Brooks, R. (1990). Elephants don't play chess. *Robotics and Autonomous Systems*, 6(1-2) :3–15.
- [Brooks et al., 1990] Brooks, R., Maes, P., Mataric, M., and More, G. (1990). Lunar base construction robots. *IEEE International Workshop on Intelligent Robots and Systems*, pages 389–392.
- [Brooks, 1991] Brooks, R. A. (1991). Intelligence without reason. In Myopoulos, John ; Reiter, R., editor, *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 569–595, Sydney, Australia. Morgan Kaufmann.
- [Bura et al., 1991] Bura, S., Drogoul, A., Ferber, J., and Jacopin, E. (1991). Eco-résolution : un modèle de résolution par interactions. In *Huitième congrès sur la reconnaissance des formes et l'intelligence artificielle (RFIA '91)*, Lyon. AFCET.
- [Cai et al., 1995] Cai, A., Fukuda, T., Arai, F., Ueyama, T., and Sakai, A. (1995). Hierarchical control architecture for cellular robotic system -simulation and experimentations-. In *IEEE International Conference on Robotics and Automation*, pages 1191–1196, Nagoya (Japan).
- [Chapelle, 2001] Chapelle, J. (2001). *Apprentissage de la coopération dans les systèmes multi-agents situés hétérogènes*. Rapport de DEA Université de Montpellier II, LIRMM.
- [Clark et al., 1992] Clark, R., Arkin, R., and Ram, A. (1992). Learning momentum : On-line performance enhancement for reactive systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 111–116, Nice, France.

- [Cohen et al., 1998] Cohen, J., Fraigniaud, P., Konig, J. C., and Raspaud, A. (1998). Optimized broadcasting and multicasting protocols in cut-through routed networks. *IEEE Transactions on Parallel and Distributed Systems*, 9 :788–802.
- [Collinot et al., 1995] Collinot, A., Carle, P., and Zeghal, K. (1995). Cassiopeia : a method for designing computational organizations. In *Proceedings of the First International Workshop on Decentralized Intelligent Multi-Agent Systems DIMAS-95*, pages 124–131.
- [Connolly and Grupen, 1992] Connolly, C. and Grupen, R. (1992). The applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7) :931–946.
- [Crowther et al., 1973] Crowther, J., Rettberg, W., Walden, R., Ornstein, D., and Heart, F. (1973). A system for broadcast communication : Reservation-aloha. In *Proceedings Sixth Hawaiï International Conference System Science*, pages 371–374.
- [de Meneses and Carmona, 1999] de Meneses, Y. L. and Carmona, O. (1999). A stereo vision system for the khepera. In *Proceedings of the 1st Intl Khepera Workshop*, Paderborn.
- [de Rosnay, 1977] de Rosnay, J. (1977). *Le microscope - vers une vision globale*. Points Essais.
- [Deneubourg et al., 1986] Deneubourg, J., Aron, S., Goss, S., Pasteels, J. M., and Duenckel, G. (1986). Random behaviour, amplification processes and number of participants : How they contribute to the foraging properties of ants. *Physica*, 22(D) :176–186.
- [Deneubourg and Goss, 1989] Deneubourg, J. and Goss, S. (1989). Collective patterns and decision-making. *Ecology, Ethology and Evolution*, 1 :295–311.
- [Deneubourg et al., 1991] Deneubourg, J., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chretien, L. (1991). The dynamics of collective sorting robot-like ants and ant-like robots. *First International Conference On the Simulation Of Adaptive Behavior (SAB 91)*, pages 356–363.
- [Doncker et al., 1998] Doncker, S., Regnier, S., and Duhaut, D. (1998). Collective path generation without marking the environment. In *Proceedings of IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pages 102–107, Victoria, B.C., Canada.
- [Dreyfus, 1979] Dreyfus, H. (1979). *Wath computers can't do : the limits of artificial intelligence*. Harper and Row.
- [Drogoul, 1993] Drogoul, A. (1993). *De la simulation multi-agent à la resolution collective de problèmes. Une étude de l'emergence de structures d'organisation dans les systèmes multi-agents*. Thèse de l'université Paris 6.

- [Drogoul and Ferber, 1992] Drogoul, A. and Ferber, J. (1992). From tom thumb to the dockers : Some experiments with foraging robots. In *2nd Int. Conf. On Simulation of Adaptative behavior*, pages 451–459, Honolulu.
- [Drogoul and Zucker, 1998] Drogoul, A. and Zucker, J. (1998). Methodological issues for designing multi-agent systems with machine learning techniques : Capitalizing experiences from the robocup challenge. In *Technical Report of the LIP6 n° 041*.
- [Durfee, 2000] Durfee, E. H. (2000). Distributed problem solving and planning. In Weiss, G., editor, *Multiagent Systems A modern approach to distributed artificial intelligence (chapter 3)*, pages 121–164. The MIT Press.
- [Durfee and Lesser, 1991] Durfee, E. H. and Lesser, V. R. (1991). Partial global planning : A coordination framework for distributed hypothesis. *IEEE Transactions on Systems, Man and Cybernetics*, 21(5) :1167–1183.
- [Elson and Estrin, 2001] Elson, J. and Estrin, D. (2001). Time synchronization for wireless sensor networks. In *Proceedings of the International Parallel and distributed Processing Symposium (IDPS), Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, San Fransisco, CA, USA.
- [Ferber, 1989] Ferber, J. (1989). *Objets et agents : une étude des structures de représentation et de communications en intelligence artificielle*. Thèse d’Etat, Université Paris 6.
- [Ferber, 1995] Ferber, J. (1995). *Les Systèmes Multi-Agents, Vers une intelligence collective, InterEditions, or Multi-Agent Systems, an introduction to Distributed Artificial Intelligenc, Addison-Wesley, 1999*.
- [Fontan and Mataric, 1996] Fontan, M. S. and Mataric, M. J. (1996). A study of territoriality : The role of critical mass in adaptive task division. In Pattie Maes, Maja Mataric, J.-A. M. J. P. and Stewart W. Wilson, eds, M. P. B., editors, *4th International Conference On the Simulation Of Adaptive Behavior (SAB 96)*, pages 553–561.
- [Fukuda et al., 1995] Fukuda, T., Nakagawa, S., Kawauchi, Y., and Buss, M. (1995). Structure decision for self organising robots based on cell structures-cebot. In *IEEE International Conference on Robotics and Automation*, pages 695–70, Scottsdale, AZ.
- [Gaudry et al., 1995] Gaudry, O., Pierrot, F., Dombre, E., and Liégeois, A. (1995). Design of a manipulator for planetary rovers. In *Proc 10th Int. Symposium RoManSy, N° 361 CISM Courses*, pages 261–267. Springer-Verlag.

- [Goldberg and Mataric, 1999] Goldberg, D. and Mataric, M. J. (1999). Coordinating mobile robot group behavior using a model of interaction dynamics. In *Proceedings, Autonomous Agents '99*, pages 100–107, Seattle, WA.
- [Goss and Deneubourg, 1991] Goss, S. and Deneubourg, J. (1991). Harvesting by a group of robots. In *Proceedings of the First European Conference on Artificial Life Conference (ECAL91)*, pages 195–204. F. Varela and P. Bourguine (Eds) MIT-Press.
- [Gutknecht, 2001] Gutknecht, O. (2001). *Proposition d'un modèle organisationnel générique de systèmes multi-agents et examen de ses conséquences formelles, implémentaires et méthodologiques*. Thèse de l'université Montpellier II.
- [Gutknecht and Ferber, 2001] Gutknecht, O. and Ferber, J. (2001). Madkit. <http://www.madkit.org/>.
- [Hadouaj et al., 2001] Hadouaj, S., Drogoul, A., and Espié, S. (2001). How to combine reactivity and anticipation : the case of conflicts resolution in a simulated road traffic. In *2n International Workshop on Multi Agent Based Simulation MABS'2000*, Boston, USA. Springer Verlag LNAI series.
- [Haninarayan and Lumelsky, 1994] Haninarayan, K. R. and Lumelsky, V. J. (1994). Sensor-based motion planning for multiple mobile robots in an uncertain environment. In *IEEE International Workshop on Intelligent Robots and Systems*, pages 1485–1492, Munich (Germany).
- [Hogg et al., 1991] Hogg, D., Martin, F., and Resnick, M. (1991). *Braitenberg Creatures*. MIT Epistemology and Learning Memorandum No. 13, Cambridge, MA.
- [Horswill, 1995] Horswill, I. (1995). Analysis of adaptation and environment. *Artificial Intelligence*, 73(1) :1–30.
- [Hull, 1943] Hull, C. (1943). *Principles of Behavior*. New York : Appleton-Century-Crofts.
- [Kaga et al., 2000] Kaga, T., Starke, J., Molnar, P., Schanz, M., and Fukuda, T. (2000). Dynamic robot-target assignment - dependence of recovering from breakdowns on the speed of the selection process. In *Distributed autonomous Robotic Systems 4*, pages 325–334. L.E. Parker, G. Bekey and J. Barhen - Springer.
- [Khatib, 1985] Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 500–505, St. Louis.
- [Khepera, ] Khepera. Miniature mobile robot, <http://www.k-team.com/robots/khepera/index.html>.

- [Koren and Borenstein, 1991] Koren, Y. and Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 1398–1404, Sacramento, California.
- [Kube and Bonabeau, 2000] Kube, C. R. and Bonabeau, E. (2000). Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, 30, Issue 1-2 :85–101.
- [Kube and Zhang, 1992] Kube, C. R. and Zhang, H. (1992). Collective robotic intelligence. In *2d International Conference On the Simulation Of Adaptive Behavior (SAB 92)*, pages 460–68, Honolulu, HI. MIT Press, Cambridge, MA.
- [Lucidarme, 2001] Lucidarme, P. (2001). <http://www.lirmm.fr/~lucidarm>.
- [Lucidarme et al., 2001] Lucidarme, P., Rongier, P., and Liégeois, A. (2001). Implementation and evaluation of a reactive multi-robot system. In *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM'01)*, pages 165–170, Como.
- [Lucidarme et al., 2002] Lucidarme, P., Simonin, O., and Liégeois, A. (2002). Implementation and evaluation of a satisfaction/altruism based architecture for multi-robot systems. In *International Conference of Robotics and Automation (à paraître)*, Washington.
- [Maes, 1991] Maes, P. (1991). A bottom-up mechanism for behavior selection in an artificial creature. In *First International Conference On the Simulation Of Adaptive Behavior (SAB 91)*, pages 238–246, Paris.
- [Maes, 1995] Maes, P. (1995). Artificial life meets entertainment : Life like autonomous agents. *Communications of the ACM*, 38, 11 :108–114.
- [Maslow, 1943] Maslow, A. H. (1943). A theory of human motivation. *Psychological Review*, 50 :370–396.
- [Mataric, 1994a] Mataric, M. J. (1994a). *Interaction and intelligent behavior*. MIT EECS PhD Thesis, MIT AI Lab.
- [Mataric, 1994b] Mataric, M. J. (1994b). Reward functions for accelerated learning. In Cohen, W. and Hirsh, H., editors, *Machine Learning Proc. of the Eleventh International Conference*, pages 181–189. Morgan Kaufmann.
- [Mataric, 1995] Mataric, M. J. (1995). Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, 16 n 2-4 :321–331.
- [Mataric, 1997] Mataric, M. J. (1997). Behavior-based control : Examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence*, 9, 2-3 :323–336.

- [Mataric, 1998] Mataric, M. J. (1998). Using communication to reduce locality in distributed multi-agent learning. *Journal of Experimental and Theoretical Artificial Intelligence, special issue on Learning in DAI Systems*, 10(3) :357–369.
- [Mataric, 2001] Mataric, M. J. (2001). Principled and efficient methods for control and learning in robot teams and humanoids. In *9th International Symposium on Intelligent Robotic Systems SIRS'2001*, Toulouse.
- [McFarland, 1987] McFarland, D. (1987). *The Oxford Companion to Animal Behaviour*. Oxford University Press.
- [Minsky, 1986] Minsky, M. (1986). *The Society of Mind*. New York : Simon and Schuster.
- [Noble, 1999] Noble, J. (1999). Cooperation, conflict and the evolution of communication. *Adaptive Behavior*, 7(3/4) :349–370.
- [Ohkawa et al., 1998] Ohkawa, K., Shibata, T., and Tanie, K. (1998). Method for controlling a group of robots by an operator. In *Distributed Autonomous Robotic Systems 3*, pages 309–318, Karlsruhe. Springer.
- [Oriolo et al., 1998] Oriolo, G., Ulivi, G., and Venditteli, M. (1998). Real-time map building and navigation for autonomous robots in unknown environments. *IEEE Transactions on systems, man and cybernetics-part B : cybernetics*, 28 no 3 :316–333.
- [Parker, 1994] Parker, L. (1994). ALLIANCE : An architecture for fault tolerant, cooperative control of heterogeneous mobile robots. In *Proceedings of the 1994 International Conference on Intelligent Robots and Systems*, pages 776–783, Munich, Germany.
- [Parker, 1995] Parker, L. (1995). L-ALLIANCE : A mechanism for adaptive action selection in heterogeneous multi-robot teams. In *ORNL/TM-13000* <http://avalon.epm.ornl.gov/~parkerle/publications/>.
- [Parker, 1999] Parker, L. (1999). Adaptive heterogeneous multi-robot teams. *Neurocomputing, special issue of NEURAP '98 : Neural Networks and Their Applications*, 28 :75–92.
- [Pirjanian, 1999] Pirjanian, P. (1999). Behavior coordination mechanisms – state-of-the-art. *Technical Report IRIS-99-375, Institute of Robotics and Intelligent Systems, School of Engineering, University of Southern California, October 1999.* <http://iris.usc.edu/~irislib>.
- [Prigogine, 1993] Prigogine, I. (1993). *Les lois du chaos*. Champs Flammarion.
- [Qutub et al., 1997] Qutub, S., Alami, R., and Ingrand, F. (1997). How to solve deadlock situations within the plan-merging paradigm for multi-robot cooperation. In *IEEE International Conference on Intelligent Robots and Systems*, volume 3, pages 1610–1615, Grenoble, France.

- [Rausch et al., 1995] Rausch, A., Oswald, N., and Levi, P. (1995). Cooperative crossing of traffic intersections in a distributed robot system. *SPIE Sensor Fusion and Networked Robotics VIII*, vol. 2589 :218–229.
- [Reynolds, 1987] Reynolds, C. (1987). Flocks, herd and schools : A distributed behavioral model. *Computer Graphics*, Vol. 21, No 4 :25–24.
- [Rohrer, 1994] Rohrer, E. (1994). timer.c <http://sac-ftp.exnet.hu/text12.html>.
- [Sen and Weiss, 2000] Sen, S. and Weiss, G. (2000). Learning in multiagent systems. In Weiss, G., editor, *Multiagent Systems A modern approach to distributed artificial intelligence (chapter 7)*, pages 259–298. The MIT Press.
- [Shibata et al., 1996] Shibata, T., Ohkawa, K., and Tanie, K. (1996). Spontaneous behavior of robots for cooperation - emotionally intelligent robot system -. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2426–2431, Minneapolis, Minnesota.
- [Simonin, 2001] Simonin, O. (2001). <http://www.lirmm.fr/~simonin>.
- [Simonin and Ferber, 2000] Simonin, O. and Ferber, J. (2000). Modeling self satisfaction and altruism to handle action selection and reactive cooperation. *6th International Conference On the Simulation Of Adaptive Behavior (SAB 2000 volume 2)*, pages 314–323.
- [Simonin and Ferber, 2001] Simonin, O. and Ferber, J. (2001). Modélisation des satisfactions personnelle et interactive d’agents situés coopératifs. In *JFIADSMA’2001, 9eme journées Francophones d’Intelligence Artificielle Distribuée et des Systèmes Multi-Agents*, Montreal.
- [Simonin et al., 2000] Simonin, O., Liegois, A., and Rongier, P. (2000). An architecture for reactive cooperation of mobile distributed robots. In *DARS 4 Distributed Autonomous Robotic Systems 4*, pages 35–44, Knoxville, TN. Springer.
- [Singh, 1998] Singh, M. P. (1998). Developing formal specifications to coordinate heterogeneous autonomous agents. In *Third International Conference on Multi-Agent Systems*, pages 261–268. IEEE.
- [SMARTTEST, 1999] SMARTTEST (1999). <http://www.its.leeds.ac.uk/projects/smartest/~index.html>.
- [Spirakis and Yap, 1984] Spirakis, P. and Yap, C. (1984). Strong NP-hardness of moving many discs. *Information Processing Letters*, 19 :55–59.
- [Steels, 1989] Steels, L. (1989). Cooperation between distributed agents through self-organization. In Demazeau, Y. and Müller, J., editors, *Decentralized AI- Proceedings of*

- the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-89)*, pages 175–196. Elsevier Science B.V., Amsterdam.
- [Steels, 1994] Steels, L. (1994). A case study in the behavior-oriented design of autonomous agents. In *Proceedings of the Third Simulation of Adaptive Behavior Conference*, pages 445–452. MIT Press, Cambridge Ma.
- [Stone and McAllester, 2001] Stone, P. and McAllester, D. (2001). An architecture for action selection in robotic soccer. In *AGENTS'01*, Montréal, Quebec. ACM.
- [Stone and Veloso, 1999] Stone, P. and Veloso, M. (1999). Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence (AIJ)*, 100 (2) :241–273.
- [Stoytchev and Arkin, 2001] Stoytchev, A. and Arkin, R. (2001). Combining deliberation, reactivity and motivation in the context of a behavior-based robot architecture. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation (IEEE CIRA 2001)*, Baniff, Canada.
- [Tanenbaum, 1998] Tanenbaum, A. S. (1998). *Computer Networks (RESEAUX 3e édition)*. InterEditions, Prentice Hall.
- [Vaughan et al., 2000] Vaughan, R., Stoey, K., Sukhatme, G., and Mataric, M. (2000). Go ahead, make my days : Robot conflict resolution by aggressive competition. In *6th International Conference On the Simulation Of Adaptive Behavior (SAB 2000)*.
- [Veloso et al., 1999] Veloso, M., Stone, P., and Bowling, M. (1999). Anticipation as a key for collaboration in a team of agents : A case study in robotic soccer. In *Proceedings of SPIE Sensor Fusion and Decentralized Control in Robotic Systems II*, volume 38 39, Boston.
- [Wang, 1994] Wang, J. (1994). On sign-board based inter-robot communication in distributed robotic systems. In *International Conference on Robotics and Automation*, pages 1045–1050. IEEE.
- [Wang, 1995a] Wang, J. (1995a). Distributed traffic regulation and control for multiple robots operating in discrete space. In *International Conference on Robotics and Automation*, pages 1619–1624. IEEE.
- [Wang, 1995b] Wang, J. (1995b). Operating primitives supporting traffic regulation and control of mobile robots under distributed robotic systems. In *International Conference on Robotics and Automation*, pages 1613–1618. IEEE.

- [Wang et al., 1995] Wang, J., Premvuti, S., and Tabbara, A. (1995). A wireless medium access protocol (csma/cd-w) for mobile robot based distributed robotic systems. In *International Conference on Robotics and Automation*, pages 2561–2566. IEEE.
- [Weiss, 2000] Weiss, G., editor (2000). *Multiagent Systems, a modern approach to distributed artificial intelligence*. G. Weiss (eds), the MIT Press.
- [Wiener, 1948] Wiener, N. (1948). *Cybernetics, or Control and Communication in Animals and Machines*. Wiley, New York.
- [Winfield and Holland, 2000] Winfield, A. and Holland, O. (2000). The application of wireless local area network technology to the control of mobile robots. *Microprocessors and Microsystems*, 23/10 :597–607.
- [Yourdon, 1989] Yourdon, E. (1989). *Modern Structured Analysis*. Englewood Cliffs, NJ : Yourdon Press, pp 139-174, 259-274.
- [Zapata et al., 1995] Zapata, R., Lépinay, P., and Deplanques, P. (1995). Collision avoidance using an egocentric memory of proximity. In *European Conference on Artificial Life ECAL 95*, pages 614–624, Granada (Spain).
- [Zeghal, 1994] Zeghal, K. (1994). *Vers une théorie de la coordination d'actions Application à la navigation aeriennne*. Thèse de l'université Paris 6.

