

LOTTO

8.1 Introduction

La série d'expériences OTTO nous a permis d'obtenir une première validation des principes du modèle en étoile et de l'algorithme de suivi d'organisation. Toutefois, ces simulations n'ont concerné qu'au maximum 16 agents, il nous a donc semblé intéressant d'étudier le comportement du modèle dans le cadre de SMA comportant un nombre beaucoup plus important d'agents. Mais l'utilisation de l'algorithme de suivi « tel quel » implique pour chaque agent de suivre l'évolution de la situation de tous les autres, ce qui pose des problèmes de mémoire avec l'augmentation de l'effectif. De plus, avec OTTO, les conditions de communication sont simulées d'une manière totalement aléatoire qui ne permet pas le « passage à l'échelle » de la simulation.

Nous avons donc conduit une nouvelle série d'expériences nommées LOTTO (*Large teams in OTTO*). Ce cadre expérimental reprend une grande partie des acquis d'OTTO en modifiant (1) la gestion de la mémoire des agents, en leur permettant de ne garder qu'une partie des informations concernant les autres membres, et (2) la gestion des possibilités de communication, en faisant se déplacer les agents dans l'espace et en leur donnant une portée de communication limitée. Notre hypothèse est que des agents ayant une capacité mémoire limitée peuvent tout de même utiliser le modèle en étoile au sein d'équipes de grande taille sans que cela ait d'effet notable sur la cohésion et la cohérence de l'équipe, s'ils sélectionnent convenablement l'information à retenir (c'est-à-dire à quels agents s'intéresser).

8.2 Principes de LOTTO

8.2.1 Mémoire

Nous avons vu que les structures de mémoire utilisées dans OTTO sont de la forme

$$S_j = (\gamma_j, D_j, \{(\bar{D}_j^1, T_1) \dots (\bar{D}_j^N, T_N)\})$$

où γ_j est le seul groupe auquel il est possible que A_j appartienne (en tout cas d'après les informations disponibles) auquel cas on a $A_j \in \gamma_j^{D_j}$ et où (\bar{D}_j^k, T_k) indique que l'agent A_j n'était pas membre du groupe $[A_k, T_k]$ à la date \bar{D}_j^k . Le groupe $[A_k, T_k]$ est (s'il existe) le groupe le plus récent que A_k ait créé et \bar{D}_j^k la date la plus récente à laquelle on sait que A_j n'était pas membre de ce groupe (toujours d'après les informations disponibles). Ces structures S_j possèdent autant d'éléments de la forme (\bar{D}_j^k, T_k) que d'agents dans l'équipe. Dans OTTO, chaque agent conserve et met à jour une de ces structures pour chacun des autres membres de l'équipe. On voit donc que l'espace mémoire nécessaire croît avec le carré de l'effectif de l'équipe.

L'idée clef derrière la gestion de la mémoire dans LOTTO est de restreindre cet espace mémoire en ne gardant qu'une partie de ces structures S_j , ce qui réduit non seulement leur nombre, mais aussi leur taille car on ne garde dans S_j un élément (\bar{D}_j^k, T_k) que si la structure S_i correspondant à ce leader est en mémoire (voir figure 8.1). En effet si on décide de ne pas s'intéresser à l'agent A_i il est inutile de garder (\bar{D}_j^i, T_i) dans les structures S_j . C'est même impossible car on ne connaît alors même plus son existence.

Dans LOTTO chaque agent ne garde donc en mémoire qu'un nombre limité N_{\max} de structures de la forme¹

$$S_j = (\gamma_j, D_j, \{(\bar{D}_j^{A_i}, T_{A_i}) \dots\})$$

pour suivre l'équipe. Au début de l'activité de l'équipe les agents ne connaissent pas sa composition, ils découvrent les autres agents présents par écoute flottante. À chaque message reçu, un agent est susceptible de découvrir l'existence d'un nouvel agent et donc de créer une structure S_j correspondante. Lorsque N_{\max} structures ont été créées et qu'un nouvel

¹Notons que les éléments de la forme $(\bar{D}_j^{A_i}, T_{A_i})$ sont plus « gros » que les (\bar{D}_j^i, T_i) de OTTO car ils doivent contenir explicitement l'identité de A_i , qui est implicite dans (\bar{D}_j^i, T_i) .

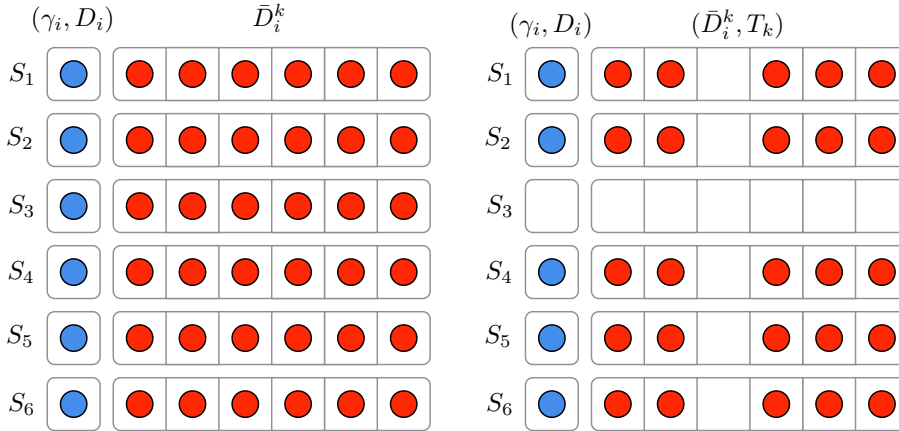


FIG. 8.1 – Utilisation comparée de la mémoire dans OTTO (à gauche) et LOTTO (à droite). Si on supprime la ligne S_3 , les éléments (\bar{D}_j^3, T_3) n'ont plus de sens dans les autres S_j car c'est comme si l'on ne connaissait plus A_3 , cela revient donc à supprimer aussi une colonne.

agent est découvert, il faut choisir entre ne pas garder trace de ce nouvel agent ou en « oublier » un autre pour lui faire de la place.

Pour déterminer quelles structures S_j garder dans l'espace mémoire limité de l'agent A_i , nous séparons ces structures par ordre d'importance en quatre classes, indiquées dans le tableau 8.1. Tout d'abord la structure la plus prioritaire pour l'agent A_i est S_i , celle qui le concerne lui-même. Ensuite viennent les structures qui concernent les agents qui appartiennent au même groupe que A_i . Cela est crucial pour préserver la cohérence intra-groupe, ne serait-ce que pour le leader. En effet, un leader *doit* connaître les membres de son groupe, la taille de la mémoire borne donc la taille des groupes possibles dans LOTTO. Ensuite viennent les leaders des autres groupes, ce sont les agents auxquels devra s'adresser A_i s'il veut rejoindre un autre groupe. Puis enfin vient le reste de l'équipe s'il reste de la place en mémoire.

Au sein d'une même classe, on donne la priorité à l'information la plus récente : S_{j_1} est préféré à S_{j_2} si $D_{j_1} > D_{j_2}$. Toutes choses étant égales par ailleurs, il semble en effet préférable de privilégier un agent dont on vient d'avoir des nouvelles par rapport à un autre dont on n'a plus entendu par-

ler depuis quelques temps et avec lequel on a donc moins de chances de pouvoir interagir.

TAB. 8.1 – Classification par ordre de priorité des structures de mémoire S_j dans LOTTO pour un agent A_i .

1	$\{S_i\}$, les informations à propos de A_i lui-même.
2	$\{S_j \mid \gamma_j = \gamma_i\}$, les autres membres du groupe de A_i .
3	$\{S_j \mid \text{lead}(\gamma_j) = A_j\}$, les leaders.
4	Le reste de l'équipe.

Cette classification nous donne un préordre total sur l'ensemble des S_j : si on a deux structures S_{j_1} et S_{j_2} , l'une est strictement préférée à l'autre ou alors elles sont équivalentes (si elles sont de la même classe et si $D_{j_1} = D_{j_2}$). Un agent A_i possède donc en mémoire \mathcal{E}_i , un ensemble de structures S_i , triées selon ce préordre, qui correspondent à sa connaissance de l'organisation de l'équipe. Lorsqu'il apprend l'existence d'un agent par écoute flottante (c'est-à-dire qu'il est fait mention dans un message d'un agent A_k et que A_i ne possède pas de structure S_k) l'agent A_i construit la structure S_k correspondante et la compare avec la structure S_{j_0} classée dernière dans \mathcal{E}_i (ou une des dernières s'il y a équivalence, le choix pouvant se faire au hasard ou selon tout autre critère). Si S_{j_0} est prioritaire ou a la même priorité que S_k , alors S_k est ignorée. Sinon, S_{j_0} est retirée de \mathcal{E}_i et S_k est insérée dans \mathcal{E}_i à la place qui lui revient, de manière à garder \mathcal{E}_i trié. Le choix de donner priorité à la structure S_{j_0} déjà en mémoire en cas d'équivalence est fait pour éviter des phénomènes cycliques et stabiliser l'ensemble \mathcal{E}_i .

8.2.2 Conditions de communication

La méthode utilisée dans OTTO pour simuler les conditions de communication n'est pas applicable si l'on augmente sensiblement le nombre d'agents dans les simulations. En effet si l'on considère par exemple un système comprenant 100 agents et une fiabilité de 50% (valeur par défaut du paramètre α dans OTTO qui correspond à la densité moyenne du graphe de communication), chaque agent a en moyenne une cinquantaine d'agents avec qui communiquer à tout instant. On voit bien qu'alors les agents risquent d'être saturés de messages, et si l'on porte le nombre d'agents à 1000, c'est cette fois à 500 interlocuteurs potentiels que chacun doit faire

face si la fiabilité reste la même ! La notion de fiabilité des communications ne semble donc pas être pertinente dans les SMA aux effectifs importants.

Pour LOTTO, nous avons donc choisi de placer les agents dans l'espace et de leur donner une portée de communication limitée afin de donner aux communications la propriété de localité (voir section 5.3.2). Ainsi, en faisant se déplacer les agents dans un espace donné, les possibilités de communication évoluent de manière continue en une bonne approximation de ce que pourraient rencontrer, par exemple, un ensemble de micro-robots qui exploreraient une zone donnée de manière collaborative.

Pour information, le tableau 8.2 donne une correspondance entre portée des communications et densité du graphe de communication (équivalente à la fiabilité dans OTTO). Bien qu'une portée de 500m semble être une valeur plutôt élevée de la portée des communications des agents si on les place dans une zone de 1km^2 , cela correspond à la fiabilité de référence dans OTTO.

TAB. 8.2 – Correspondance entre portée des communications et densité moyenne du graphe de communication pour 50 agents répartis dans un carré de 1km de côté.

Portée	densité du graphe
100m	3%
300m	22%
500m	49%

8.3 Protocole expérimental

Comme pour OTTO, les processus de décision de plusieurs agents sont simulés par des machines à états finis dont l'évolution suit des lois de Poisson $P(t/\tau)$ (voir section 7.2.1). À chaque état est associée une valeur temporelle qui correspond à la durée moyenne durant laquelle la machine va rester dans cet état. Les machines sont d'ailleurs les mêmes que celles qui sont utilisées pour OTTO, elles se utilisent donc les mêmes paramètres avec les mêmes valeurs par défaut.

8.3.1 Déplacements des agents

Afin d'obtenir des possibilités de communication « réalistes », nous avons choisi de faire évoluer les agents dans l'espace. Au départ d'une simulation, les agents sont répartis de manière aléatoire dans un carré de 1km^2 (un tirage avec une loi de probabilité uniforme pour les abscisses et un autre pour les ordonnées) et chacun est doté d'une orientation, elle aussi aléatoire et uniforme entre 0° et 360° .

Par la suite, les agents évoluent droit devant eux à une vitesse constante de $5\text{m}\cdot\text{s}^{-1}$ et changent de direction à des instants déterminés par une loi de Poisson de constante 10s. En moyenne toutes les dix secondes, une nouvelle direction est donc tirée aléatoirement pour chaque agent. La figure 8.2 montre des exemples de trajectoires.

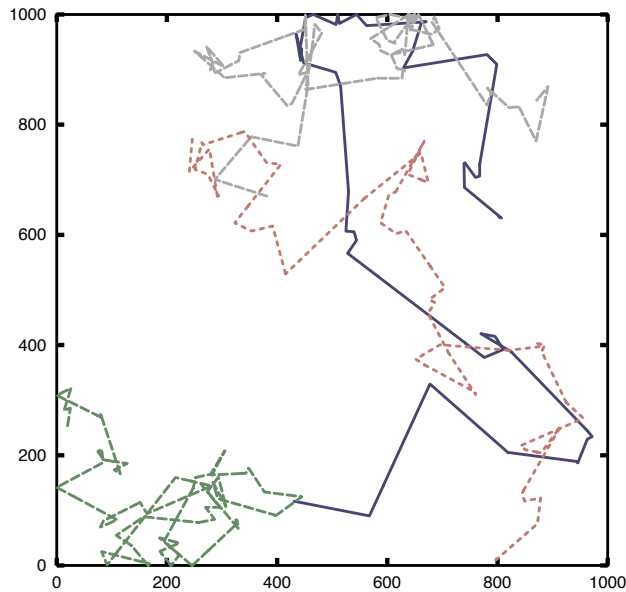


FIG. 8.2 – Trajectoires de quatre agents pendant 10 minutes

8.3.2 Paramètres

Les simulations OTTO nous ont permis d'identifier le nombre d'agents (N), la fréquence des vérifications (T_{CHK}) et la rectification (*Rectify*) comme

étant les paramètres les plus pertinents du modèle en dehors des paramètres des communications. Les caractéristiques de LOTTO nous amènent à y ajouter la portée des communications et la mémoire des agents. Ayant déjà mis en évidence le rôle crucial de la rectification, elle est mise en œuvre par défaut. Restent donc quatre paramètres à fixer pour une simulation dans LOTTO :

- N : le nombre d'agents présents dans la simulation ;
- P : la portée de communication des agents ;
- T_{CHK} : l'intervalle moyen entre deux CHK émis par un agent ;
- Mem : le rapport entre le nombre maximal de structures S_j en mémoire et N .

TAB. 8.3 – Domaines de variation des paramètres de LOTTO.

Paramètre	Domaine de variation
N	30 – 60 – 90 – 120
P	50m – 200m – 350m – 500m
T_{CHK}	12,5s – 25s – 37,5s – 50s
Mem	0,125 – 0,25 – 0,375 – 0,5 – 0,625 – 0,75 – 0,875 – 1,0

8.3.3 Mesure de performances

Tout comme avec OTTO, chaque simulation est évaluée en faisant la moyenne sur sa durée de critères calculés de manière instantanée. Deux critères sont utilisés pour évaluer le comportement de LOTTO durant des simulations de 10 minutes.

Comme avec OTTO, la *cohésion* est la proportion d'agents qui sont membres d'un groupe sans en être le leader. Pour être considéré comme membre d'un groupe, l'agent et le leader doivent le croire. Une cohésion importante indique que l'équipe est composée de quelques grands groupes tandis qu'une faible cohésion indique une équipe très fragmentée.

En revanche, les cohérences forte et faible utilisées dans OTTO ne sont pas utilisables dans LOTTO. En effet, avec ces critères définis pour OTTO une équipe est soit cohérente (fortement ou faiblement) soit incohérente, et le résultat d'une simulation correspond à la proportion du temps que passe l'équipe dans un état cohérent. Avec le grand nombre d'agents utilisés dans

LOTTO il devient très peu probable d’avoir une situation entièrement cohérente à un moment donné (cf. section 7.3.2). Plutôt que de mesurer la cohérence de l’équipe, nous choisissons donc de mesurer le nombre moyen d’incohérences par agent durant une simulation.

DÉF. 8.1 – Incohérence. Une situation est incohérente si :

- un leader croit qu’un agent est membre de son groupe sans que cela soit le cas ; ou bien inversement si
- un agent croit être membre sans que cela soit le cas.

L’incohérence d’une équipe à un instant donné est égale au nombre de tels cas ramené à un agent (c’est-à-dire divisé par N).

8.4 Résultats et analyse

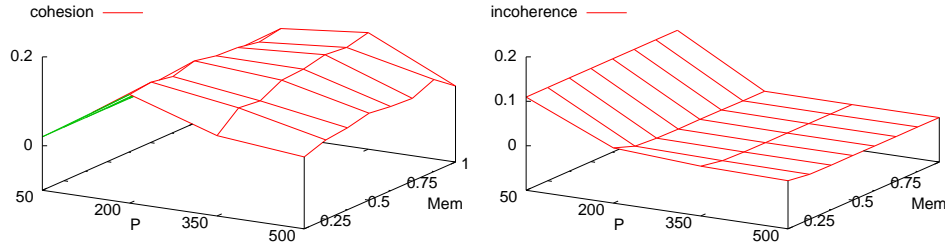
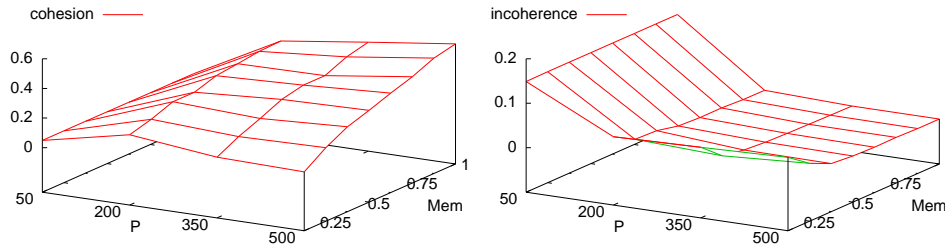
Nous présentons ici les résultats des simulations menées sur LOTTO. Ces résultats sont sous la forme de graphes en trois dimensions représentant les effets de la variation de la taille de la mémoire des agents (Mem) croisées avec la variation du nombre d’agents (N), de la portée des communications (P) ou de l’intervalle de temps entre les vérifications (T_{CHK}).

Pour chaque « point » de simulation, c’est-à-dire un quadruplet (N, Mem, P, T_{CHK}) , trois simulations de 10 minutes sont faites et on calcule la moyenne de leurs résultats (*cohésion* et *incohérence*). L’espace des paramètres de LOTTO a été exploré selon des « plans » en fixant à chaque fois deux paramètres parmi $\{N, P, T_{CHK}\}$ et en étudiant l’influence des variations croisées du troisième paramètre et de la mémoire (qui nous intéresse ici particulièrement).

8.4.1 Portée des communications

Les figures 8.3 et 8.4 montrent l’évolution de la *cohésion* et de l’*incohérence* pour différentes valeurs de P et Mem .

Considérons l’évolution de la *cohésion* sur la figure 8.3. On peut observer que, ici encore, la taille de la mémoire ne semble pas avoir d’influence. En revanche, la portée des communication a ici une influence particulière, il existe – avec les valeurs de paramètres de cette série de simulations – une valeur de la portée qui semble optimale pour la cohésion ($P \simeq 200m$).

FIG. 8.3 – Cohésion et incohérence pour $N = 120$ et $T_{CHK} = 5s$.FIG. 8.4 – Cohésion et incohérence pour $N = 120$ et $T_{CHK} = 30s$.

On peut interpréter ceci en remarquant qu'aussi bien un manque de possibilités de communication qu'un surcroît de messages sont nuisibles à la *cohésion*.

Sur la figure 8.4 en revanche, on peut observer l'influence croisée de la taille de la mémoire et de la portée des communications sur la *cohésion*. La *cohésion* augmente avec chacun de ces deux paramètres. L'augmentation de la portée permet à plus d'agents d'interagir ce qui fait qu'ils peuvent donc former de plus gros groupes, la mémoire quant à elle limite la taille maximale que peut avoir un groupe, son augmentation permet donc également d'augmenter la *cohésion*, en permettant aussi aux agents de conserver des informations à propos de plus de leaders et donc d'augmenter leurs chances de rejoindre un groupe.

En ce qui concerne l'*incohérence*, on peut voir que la taille de la mémoire n'a pour ainsi dire aucune influence sur sa valeur. On ne note un effet que si la mémoire est très réduite et les vérifications espacées dans le temps

($T_{CHK} = 30s$). Il s'agit là de deux facteurs favorisant *a priori* l'apparition des incohérences, et ils doivent se conjuguer pour avoir un impact même faible. En revanche, on note sur les deux figures un effet notable de la portée des communications sur l'*incohérence*. L'augmentation de P réduit l'apparition des incohérences en augmentant les possibilités de communication des agents, et donc leur accès à l'information.

8.4.2 Nombre d'agents

La figure 8.5 illustre l'évolution de la *cohésion* et de l'*incohérence* pour différentes valeurs de N et de Mem , avec $P = 100m$ et $T_{CHK} = 50s$. On peut noter que la taille de la mémoire n'a d'effet ni sur la *cohésion*, ni sur l'*incohérence*. La seule variation observable est l'augmentation générale de la *cohésion* avec la taille de l'équipe (N) : avec une portée limitée de 100m une augmentation de N augmente les possibilités de communication des agents et donc également la *cohésion*.

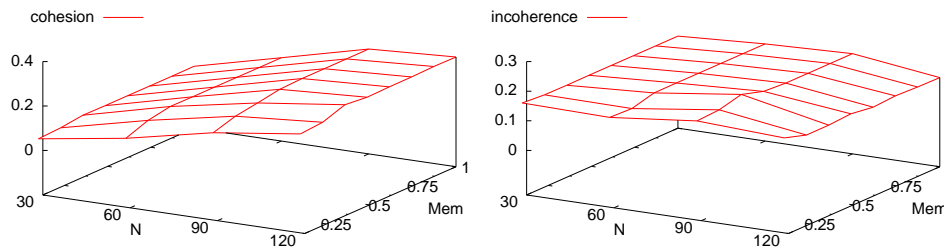


FIG. 8.5 – Cohésion et incohérence pour $P = 100m$ et $T_{CHK} = 50s$.

La figure 8.6 illustre l'évolution de la *cohésion* et de l'*incohérence* pour différentes valeurs de N et de Mem , avec $P = 500m$ et $T_{CHK} = 50s$. D'une part, la grande portée des communications fait que l'augmentation du nombre d'agents n'améliore pas autant les communications qu'à la figure 8.5, donc N n'a que très peu d'effet sur la *cohésion*. D'autre part, avec de si bonnes conditions de communication, les agents peuvent tirer parti d'une mémoire étendue pour créer de grands groupes et donc obtenir une bonne *cohésion*, ce qui explique l'augmentation régulière de la *cohésion* avec la mémoire.

En ce qui concerne l'*incohérence*, une mémoire réduite conduit à des incohérences car les agents peuvent interagir avec trop d'autres membres de

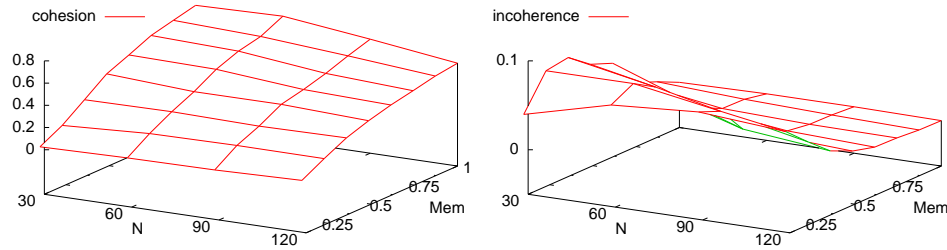


FIG. 8.6 – Cohésion et incohérence pour $P = 500\text{m}$ et $T_{CHK} = 50\text{s}$.

l'équipe et perdent certaines informations. D'une certaine manière, la composition de la mémoire change trop rapidement, cela pourrait sans doute être évité si les agents avaient des processus de décision plus évolués que de simples machines à états finis.

8.4.3 Fréquence des vérifications

Les figures 8.7 et 8.8 illustrent l'évolution de la *cohésion* et de l'*incohérence* avec T_{CHK} et Mem . Nous pouvons observer que de manière générale, T_{CHK} n'a pas d'impact notable ni sur la *cohésion* ni sur l'*incohérence*. Ceci est tout à fait en accord avec les résultats obtenu avec OTTO (qui correspond à LOTTO avec $Mem = 1$) au chapitre 7 et avec la conjecture selon laquelle, grâce à l'écoute flottante, il est possible d'obtenir une bonne cohésion sans que les membres de l'équipe vérifient sans cesse leur statut.

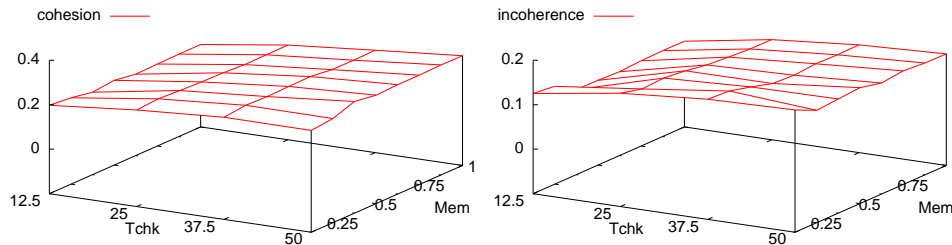


FIG. 8.7 – Cohésion et incohérence pour $P = 100\text{m}$ et $N = 120$.

Comme plus haut avec les variations de N , la mémoire n'a d'effet que lorsque les agents ont de très bonnes conditions de communication, ce qui correspond à la figure 8.8 (120 agents avec une portée de 500m). Une mémoire limitée apporte des incohérences car nos agents sont « idiots » et interagissent avec plus d'agents qu'ils ne peuvent se rappeler effectivement. Et enfin, une bonne mémoire permet une bonne *cohésion*.

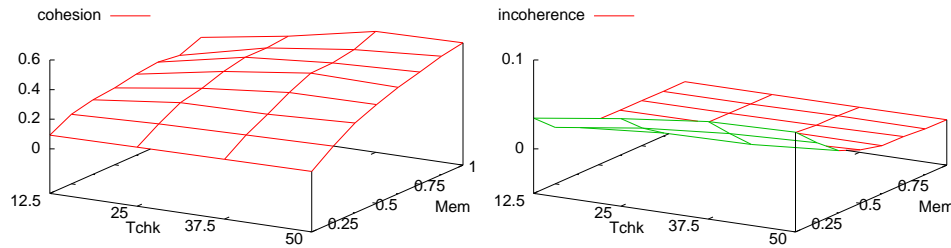


FIG. 8.8 – Cohésion et incohérence pour $P = 500\text{m}$ et $N = 120$.

8.4.4 Discussion générale

À partir de ces expériences, nous pouvons conclure du comportement de LOTTO que limiter la taille de la mémoire des agents n'a un impact négatif sur l'*incohérence* au sein de l'équipe et ne limite sa *cohésion* que dans le cas où les agents ont beaucoup de facilités de communication (densité importante des agents ou grande portée des communications). On notera tout de même que les effets de la limitation en mémoire sont plus visibles sur la *cohésion* que sur l'*incohérence*.

Or, LOTTO a pour objectif de fonctionner avec de mauvaises conditions de communication, et dans ce cadre, une mémoire même très réduite n'affecte en rien la *cohésion* ni l'*incohérence* par rapport à ce qui serait obtenu avec OTTO. De plus, il faut noter que les agents utilisés dans ces simulations n'étaient que de très simples machines à états finis, et qu'une équipe d'agents aux capacités de décision plus élaborées obtiendrait de bien meilleures performances, notamment en évitant la confusion qui résulte du manque de mémoire face à trop d'interactions.

8.5 Conclusion

Les simulations LOTTO ont permis de montrer qu'il est possible d'utiliser les principes du modèle en étoile évalués dans OTTO tout en fixant une limite à la mémoire utilisée par chaque agent pour stocker ses informations à propos de l'équipe. Des simulations ont montré que la stratégie utilisée dans LOTTO pour classer les informations permet à chaque agent de ne garder que les informations les plus pertinentes pour lui, d'oublier le reste, et cela sans effet notable sur la *cohésion* et l'*incohérence* de l'équipe, à condition qu'elle évolue dans des conditions « normales » d'utilisation de LOTTO qui sont : mauvaises possibilités de communication entre agents et mémoire limitée.

Conclusion générale

Dans la première partie de ce manuscrit (chapitres 1 et 2), nous avons exposé les fondements et le cadre de notre travail. Il existe une grande variété d'approches pour la conception d'agents artificiels et de systèmes multiagents, et la plupart d'entre elles traitent des structures qui gouvernent les interactions entre agents (ou qui en émergent). Ces structures n'ont pas d'existence physique, elles n'existent que « dans la tête » des agents, plus précisément il s'agit de régularités entre les informations dont disposent plusieurs agents.

Ces informations dont dispose un agent peuvent concerner directement d'autres agents (il peut alors les prendre en compte pour atteindre ses objectifs), la manière dont il peut et doit interagir avec les autres (des règles sociales), ou encore simplement comment il doit déterminer son propre comportement (c'est alors *via* l'environnement et le comportement des agents que les structures émergent).

Si l'on considère ainsi que les structures des SMA résultent de régularités entre les informations dont disposent les agents, la communication est alors un type d'interaction bien particulier, et la deuxième partie du manuscrit lui est consacrée (chapitres 3 et 4). En effet, la communication a justement pour objet de transférer de l'information entre les agents. On voit donc bien à quel point la communication est fondamentale pour qu'un système multiagent atteigne et conserve une (la) structure qui lui permette de remplir sa fonction. Mais transférer ainsi de l'information d'agent en agent est loin d'être trivial. Il faut tout d'abord disposer d'un médium, d'un aspect de l'environnement qui serve de support au signal. Ensuite, il faut que les deux agents possèdent les mêmes fonctions de codage et décodage du signal, que l'on retrouve dans la théorie classique de la communication.

Mais ceci ne permet d'accéder qu'aux niveaux de communication les

plus simples. D'une part la communication doit être considérée non pas comme le simple transfert d'un signal entre deux agents, mais comme une activité conjointe qui nécessite un certain nombre de connaissances communes entre les interlocuteurs (règles de conversation, correspondance entre symboles et environnement, *etc.*) Mais d'autre part la communication est très souvent considérée selon une approche dyadique et abstraite assez rigide, alors que nous voulons justement nous en servir dans des groupes d'agents évoluant dans un environnement réel. Ceci nous a donc amené à faire l'hypothèse qu'en s'inspirant de la communication humaine, il est profitable d'utiliser le phénomène de *l'écoute flottante* dans des systèmes multiagents afin de donner souplesse et robustesse à leur structure.

Pour aller dans ce sens, dans une troisième partie (chapitres 5 et 6) nous avons présenté le *modèle en étoile*. Il permet aux membres d'un SMA d'utiliser l'écoute flottante pour se structurer en groupes centrés sur des leaders afin de s'adapter aux conditions de communication limitées et dynamiques dont ils disposent. Toutefois ce modèle est normatif, il fournit un cadre dans lequel les agents peuvent évoluer, il ne dicte pas comment et pourquoi les agents prennent leurs décisions, mais assure que si leurs décisions respectent certaines contraintes (notamment de cohérence) ils pourront tirer parti de tous les messages qu'ils pourront recevoir (même s'ils ne leur sont pas destinés) afin de se forger les croyances les plus cohérentes possibles sur l'organisation de l'équipe et ainsi faciliter son activité.

Nous avons proposé plusieurs éléments utiles pour exploiter ce modèle en étoile. Tout d'abord nous avons présenté un langage qui permet aux agents de communiquer à propos de l'organisation de l'équipe en groupes. Ce langage est fondé sur un ensemble de six types d'*attitudes propositionnelles* dont les sémantiques sont définies formellement. Nous avons ensuite présenté un mécanisme qui permet à un agent de tenir à jour ses croyances à propos de l'organisation de son équipe, en fonction de messages exprimés dans ce langage et qu'il reçoit par écoute flottante. Nous disposons également d'un algorithme qui permet de reproduire ce mécanisme de manière incrémentale, indépendamment de l'ordre d'arrivée des messages, en temps constant et en utilisant un espace mémoire de taille fixe (pour une équipe donnée).

Dans une quatrième partie (chapitres 7 et 8), nous avons présenté les résultats de deux séries d'expériences menées sur le modèle en étoile en

simulation. La première série, nommée OTTO, nous a permis de valider expérimentalement les principes du modèle. Nous avons pu constater que conformément à nos hypothèses, l'utilisation de l'écoute flottante permet à une équipe d'adapter son organisation dynamiquement en fonction des conditions de communication. Mais elle permet aussi des comportements d'une grande souplesse tels que la rectification « spontanée », et ce sans l'explosion du nombre de messages que la souplesse d'utilisation des communications aurait pu nous faire craindre.

Après avoir mis à l'épreuve nos hypothèses avec un nombre limité d'agents, la deuxième série de simulations nommée LOTTO nous a permis de confronter le modèle en étoile à des systèmes d'une autre échelle, toujours en simulation. Les résultats ont montré que le modèle supporte le « passage à l'échelle » et même alors que des contraintes de mémoire supplémentaires sont imposées aux agents.

Ce travail se veut une étape dans le développement de l'utilisation de modèles de communication alternatifs tels que l'écoute flottante dans le domaine des systèmes multiagents. S'inspirer du fait humain dans le domaine de l'intelligence artificielle n'est pas nouveau, ainsi, d'après Kraus [Kraus, 1997], « *Dans les environnements non structurés et imprévisibles, il peut être utile d'utiliser des heuristiques pour la coopération et la coordination entre agents autonomes, basées sur des techniques humaines efficaces de coopération et d'interaction.* » Mais étrangement, et même si une tendance semble s'amorcer actuellement dans la communauté, peu de travaux allant dans ce sens ont été conduits dans le domaine de la communication entre agents artificiels.

Nous avons avant tout proposé et mis à l'épreuve expérimentalement l'hypothèse que l'écoute flottante peut permettre des modes de coopération originaux, souples et efficaces dans les systèmes multiagents. Cette hypothèse s'accompagne d'un modèle normatif qui permet de structurer dynamiquement et pour un faible coût une équipe en fonction de ses possibilités de communication.

Le déploiement de groupes de drones ou de micro-drones constituerait une application idéale pour le modèle en étoile, et a d'ailleurs servi de motivation tout au long de nos travaux. Les caractéristiques de faible coût en puissance de calcul et en mémoire du mécanisme de suivi d'organisation le rendent séduisant pour de telles applications, qui comprennent un

grand nombre d'engins individuellement assez simples et qui opéreraient dans des conditions très difficiles. La présence de leaders dans le modèle constitue lui aussi un avantage, mais cette fois du point de vue de l'interface homme-machine. Si l'on envisage plusieurs dizaines d'engins opérant dans une zone, pour accomplir une tâche donnée, se pose le problème du contrôle par un opérateur. En effet, si nous voulons des machines capables de plus en plus d'autonomie, nous voulons aussi pouvoir les contrôler, et les leaders pourraient jouer ici un rôle d'intermédiaires tout trouvé entre un opérateur humain et les membres de leur groupes. En effet, il est difficile d'imaginer l'opérateur donner des ordres individuellement à chaque robot.

D'un point de vue plus théorique, il serait intéressant d'étudier quels types de modèles pourraient être développés sans la notion de leader pour représenter un groupe et « centraliser » les informations. Le fait de disposer de leaders nous a permis d'avoir une définition en intension d'un groupe, est-ce possible sans cette sorte d'« ancrage physique » ? Peut-être pourrions-nous imaginer un groupe simplement comme une sorte de cooptation de ses différents membres, mais alors comment l'obtenir ? Et comment obtenir les connaissances à partir des désirs et croyances des agents ? Une piste pourrait être de considérer qu'un agent est (était) membre à la date qui correspond au minimum des dates auxquelles chacun des autres le croit membre. Tout ceci pose des problèmes intéressants de cohérence des croyances individuelles.

Notre modèle traite des connaissances du passé (il permet même de les modifier !), mais nous n'avons pas abordé le futur. Pourtant le fait de disposer ainsi de connaissances et de désirs datés pourrait permettre de tirer parti de certains travaux liés à l'engagement (*commitment*). Par exemple, si l'on décrète que le fait d'exprimer le désir d'être membre à une date t implique de s'engager à le conserver pendant un intervalle de temps $[t, t + \delta]$ un leader pourrait affirmer l'appartenance *immédiate* d'un agent à son groupe s'il reçoit son JOIN durant cet intervalle.

Enfin, pour revenir aux activités humaines, il pourrait être intéressant d'étudier à quel point le type de modèle que nous proposons peut se comparer à la notion de *co-présence*. En effet, le fait d'être en présence de son interlocuteur et d'avoir en continu un « retour » même non verbal sur les communications que l'on échange avec lui, constitue un des fondements de la communication humaine et il est probable qu'un équivalent serait précieux pour des agents artificiels.

Annexe A

STEAM

La partie des travaux de Milind Tambe auxquels nous nous intéressons ici se situe dans le cadre des environnements de simulation interactifs. En l'occurrence il s'agit de concevoir des pilotes artificiels pour des simulations de champs de bataille dans une optique d'entraînement. La première étape de ce développement est TacAir-Soar [Tambe *et al.*, 1995], des agents développés à partir de 1992 sur l'architecture Soar dans l'optique d'être utilisés pour les exercices militaires STOW-E (*Simulated Theater Of War, Europe*) puis STOW-97.

Ces exercices sont basés sur le DIS (*Distributed Interactive Simulation*) [DIS Steering Committee, 1994]. Il s'agit d'un environnement de simulation en réseau permettant de faire interagir de nombreuses entités dans un même monde virtuel. Par exemple, STOW-E (qui a eu lieu du 4 au 7 novembre 1994) implique environ 2000 entités telles que des avions, des bateaux, des véhicules terrestres et même des fantassins. Certaines de ces entités correspondent à de véritables engins transmettant leur statut sur le réseau DIS (par exemple des chasseurs-bombardiers F-18) ou à des simulateurs pilotés par des humains. Toutefois, la majorité de ces entités sont contrôlées par des ordinateurs communiquant avec le DIS par l'intermédiaire d'interfaces semblables, sur le principe, aux simulateurs proposés aux opérateurs humains et auxquels on impose des contraintes humaines (résistance à l'accélération, temps de réaction).

Au cours des quatre jours d'exercice de STOW-E, les agents TacAir-Soar remplissent des missions aériennes assez diverses (avec un certain succès), qui permettent d'éprouver le système et de mettre en évidence les directions à prendre pour les développements ultérieurs. Plusieurs critères et