

# Approche distribuée et auto-adaptative pour la régulation de la taille d'une population d'agents mobiles dans un réseau

M. Bakhouya<sup>1</sup> and J. Gaber<sup>2</sup>

<sup>1</sup>HPCL, George Washington University, Washington DC, USA

<sup>2</sup>Université de Technologie de Belfort-Montbéliard, 90010 Belfort Cedex, France  
bakhouya@gwu.edu, gaber@utbm.fr

**Résumé**—Dans cet article, une approche distribuée et adaptative inspirée du système immunitaire pour la régulation d'une population d'agents mobiles pouvant se cloner dans un réseau est présentée. Il s'agit d'une approche fondée sur l'utilisation d'un modèle de comportement d'agents, dans lequel un mécanisme de décision local leur permet de sélectionner le comportement le plus approprié en fonction de l'état de leur environnement sans recourir à des paramètres définis d'une manière globale ou fixés à priori. La régulation de la taille de la population émerge, d'une manière globale, comme résultante des actions locales des agents autonomes distribués dans le réseau.

**Mots clés**—Réseau à grande échelle, agents mobiles, clonage, régulation, système immunitaire, colonie de fourmis.

## I. INTRODUCTION

Le paradigme d'agents mobiles a été introduit pour la première fois par White [13] dans le cadre de la mise en oeuvre d'un langage orienté objet appelé Telescript pour le développement d'applications distribuées. Un agent mobile est un processus que l'on peut faire migrer et exécuter sur les différents sites d'un réseau pour réaliser une tâche pour le compte d'un utilisateur ou d'une application [2,12]. Il est constitué de trois parties : les données, le code et un contexte d'exécution [11]. Parmi les données d'un agent, on trouve son identifiant et éventuellement son itinéraire [26]. Le code de l'agent représente le programme de la tâche qui lui a été assignée. Le contexte d'exécution constitue l'état d'exécution courant de l'agent mobile.

Pour faire migrer et exécuter un agent mobile sur les différents sites d'un réseau, un environnement d'exécution d'agents mobiles tel que Aglets [26] est nécessaire. Il s'agit d'une plateforme qui fournit une interface de programmation et d'exécution offrant plusieurs services et primitives de base. Parmi ces dernières, on trouve celles permettant la création et le clonage, l'activation et la migration des agents ainsi que les services de communication inter agents [26]. Plus précisément, un environnement d'exécution d'agents mobiles

est constitué d'un ensemble de programmes, appelés places, s'exécutant sur les sites susceptibles d'accueillir des agents. Les places sont des programmes qui fournissent aux agents l'infrastructure de base pour leur exécution et leur permet de se déplacer.

Ce paradigme d'agents mobiles engendre aujourd'hui une activité de recherche très importante dans le domaine des réseaux et des systèmes distribués [7,8]. En particulier, il est considéré comme l'alternative au paradigme client/serveur pour la recherche d'information [9,12], l'accès aux données réparties dans un réseau [10] et pour l'exécution de tâches à distance et la gestion du réseau [11]. De plus, le paradigme d'agents mobiles dotés de capacités d'expansion, de réduction et d'adaptation dans un environnement dynamique permet d'envisager aujourd'hui leur utilisation pour la mise en oeuvre de nouvelles approches dans le cadre de l'ubiquité informatique et l'informatique diffuse [19,20].

En effet récemment, deux nouveaux paradigmes alternatifs au paradigme d'interaction traditionnel client/serveur (CSP) ont été proposés par Gaber dans [19,20]: le premier paradigme est inverse au paradigme traditionnel client/serveur; c'est le service qui vient vers le client et non le client qui prend l'initiative et demande une ressource ou un service en connaissant à priori son existence et sa localisation. Le deuxième paradigme met en oeuvre le principe de l'auto-organisation pour permettre l'émergence spontanée de nouveaux services dans un environnement ad hoc sans aucune planification au préalable et d'une manière imprévisible. Le premier paradigme (SCP) permet la mise en oeuvre de l'ubiquité numérique (Ubiquitous computing) dans un réseau dynamique à grande échelle issue du croisement des réseaux filaires, non filaires et mobile. Le deuxième paradigme (SEP) est plus adapté à l'informatique ambiante ou diffuse (Pervasive computing) pour la mise en oeuvre de l'auto-organisation et l'émergence dans le contexte des réseaux mobiles ad hoc et des réseaux de capteurs sans fil [19,20,25,27].

Le premier paradigme a été mis en oeuvre par l'intermédiaire d'un middleware basé sur l'utilisation des

agents mobiles et inspirée des concepts du système immunitaire de l'Homme pour la découverte de services dans les réseaux à grande échelle [2,19,20,21,24]. Plus précisément, cette approche est fondée sur la construction et la suppression de communautés de serveurs en fonction des requêtes des utilisateurs et la disponibilité des ressources. Une communauté de serveurs est un réseau d'affinités liant entre eux des serveurs capables de fournir, ensemble, un service. La création d'une communauté peut s'effectuer d'une manière proactive ou réactive. Les créations proactives des communautés sont initiées par des serveurs alors que les créations réactives sont effectuées lors de la résolution des requêtes émises par des utilisateurs [2,21].

Le processus de création proactive est un processus de détection de communautés qui permet aux serveurs (i.e., nœuds pouvant offrir des ressources) de se découvrir mutuellement dans le but de créer des relations d'affinités. Ce processus de détection est basé sur l'utilisation d'agents mobiles qui exécutent des marches aléatoires parallèles pour la découverte des services présents dans le réseau. Une marche aléatoire dans un graphe  $G(n,m)$ , où  $n$  est le nombre de sommet et  $m$  est le nombre de liens, est un processus stochastique dans lequel un mobile se déplace aléatoirement entre les nœuds du graphe. Broder et Karlin dans [18] ont montré qu'un mobile effectuant une marche aléatoire dans  $G$  visite tous les sommets de  $G$  au moins une fois quand la longueur moyenne de sa marche est en  $O(n^2 \log(n))$ . Grâce à ce processus de dissémination via les agents mobiles, les nœuds disposant des ressources nécessaires pour mettre en oeuvre un service, se découvriront mutuellement et pourront fournir conjointement ce service. Plus précisément, une communauté de ces nœuds est créée pour représenter le service. Il est considéré qu'un service peut être constitué d'un ensemble de ressources éparpillées dans le réseau. Un nœud disposant d'une partie de ces ressources ne peut fournir le service d'une manière individuelle. Le temps de couverture de la marche aléatoire peut être amélioré en permettant aux agents mobiles de se cloner dans le réseau [2,17].

La résolution des requêtes est effectuée selon une stratégie de résolution gloutonne basée sur la sélection du meilleur voisin au sein d'une communauté, avec renforcement des liens d'affinités. Plus précisément, un agent mobile représentant une requête, une fois créé, initie une marche aléatoire dans le réseau jusqu'à atteindre un nœud pouvant fournir une ou plusieurs ressources recherchées pour composer le service. Si le nœud rencontré est le point d'entrée d'une communauté correspondante au service recherché, l'agent effectue un parcours glouton au sein de cette communauté en fonction des valeurs des liens d'affinités. Il emprunte un lien lorsque sa valeur d'affinité est la plus élevée. De plus, lors de son parcours, il renforce la valeur de l'affinité des liens sélectionnés et diminue celles des liens non sélectionnés [2,21]. Il s'agit d'un processus de renforcement local des liens. Lorsque l'agent termine son parcours au sein de la communauté et constitue la liste des serveurs fournissant le service, il procède au renforcement du chemin sélectionné.

Plus précisément, il emprunte le chemin à l'envers jusqu'au point d'entrée en renforçant une deuxième fois les liens choisis. Il s'agit d'un deuxième processus de renforcement global du chemin sélectionné.

D'autres approches basées sur l'utilisation des agents mobiles ont été proposées dans [22,23] pour la découverte de ressources dans les réseaux de pair à pair (P2P). Ces approches font appel à l'opération de clonage qui permet à un agent de créer une ou plusieurs copies de lui-même. En d'autres termes, cette opération permet aux agents de se dupliquer et de se multiplier.

Il est important de noter qu'une approche distribuée basée sur le clonage est difficile à mettre en oeuvre dans un réseau à grande échelle car elle soulève le problème suivant : comment réguler la taille de la population de ces agents et limiter sa croissance pour éviter la surcharge du réseau. Plus précisément, la variation de la taille de la population d'agents doit s'adapter aux ressources du réseau et ceci, d'une manière distribuée, sans s'appuyer sur une structure globale ou une information centralisée difficile voir impossible à obtenir.

Nous commençons dans la section 2 par la présentation d'une approche inspirée par la méthode de la colonie de fourmis [1] et qui a été proposée dans un cadre précis, celui de la mise en oeuvre d'un algorithme de routage par vecteur de distance en utilisant des agents. Nous présentons ensuite dans la section 3 une approche distribuée et auto-adaptative inspirée du système immunitaire de l'Homme [2,3,16]. Dans la section 4, nous présentons des résultats de simulation. La section 5 présente la conclusion et des perspectives.

## II. L'APPROCHE INSPIREE DE LA COLONIE DE FOURMIS

### A. Principe de l'approche de régulation

Amin et Mikler ont proposé dans [1] une approche, inspirée par la méthode de la colonie de fourmis, pour réguler la taille d'une population d'agents mobiles dans le cadre de la mise en oeuvre d'un algorithme de routage par vecteur de distance en utilisant des agents (Agent-based Distance Vector Routing, ADVR). Ce type de protocole permet aux routeurs de transmettre des messages contenant des vecteurs de routage à leurs voisins. Dans cette approche, les vecteurs de routage sont transportés par des agents mobiles, représentant les fourmis, qui se déplacent entre les nœuds du réseau. Plus précisément, un agent mobile se déplace d'un nœud  $i$  à un nœud  $j$  en transportant un vecteur  $R_i$  selon une stratégie de déplacement. Cette dernière définit la manière avec laquelle l'agent choisi le prochain nœud à visiter parmi les nœuds voisins de  $i$ . La stratégie utilisée par Amin et Mikler dans [1] est le déplacement aléatoire (i.e. la marche aléatoire) entre les nœuds du réseau. Une fois l'agent mobile atteint le nœud  $j$ , il communique le vecteur transporté à ce nœud, obtient un nouveau vecteur  $R_j$  et prend ensuite une décision en sélectionnant une action parmi les trois actions suivantes: migrer, cloner ou s'auto-éliminer. Plus précisément, l'approche de régulation est mise en oeuvre de la manière suivante. Chaque nœud  $i$  du réseau enregistre la date  $t_a$

correspondante à la dernière visite d'un agent mobile. Lorsqu'un nouvel agent mobile visite ce nœud à un instant  $t_b$ , il obtient de  $i$  la date  $t_a$  et calcule la valeur  $\lambda(t_b - t_a)$ . Cette valeur représente la volatilité de la phéromone,  $\lambda$  étant une constante et  $(t_b - t_a)$  représente le temps d'interarrivée de deux agents successifs au nœud. En fonction de cette valeur et de deux autres paramètres prédéfinis  $MIN$  et  $MAX$ , l'agent sélectionne une action parmi les trois actions suivantes : *cloner*, *migrer* ou *s'auto-éliminer*. Ce choix s'effectue de la manière suivante :

- si la valeur  $e^{-\lambda(t_b - t_a)}$  est au-dessus du seuil maximal  $MAX$ , alors l'agent s'autoélimine,
- si la valeur  $e^{-\lambda(t_b - t_a)}$  est en dessous du seuil minimal  $MIN$ , alors l'agent se clone,
- si la valeur  $e^{-\lambda(t_b - t_a)}$  est comprise entre le seuil  $MAX$  et le seuil  $MIN$ , alors l'agent ne se clone pas et se déplace vers un autre nœud choisi aléatoirement parmi les nœuds voisins.

Ce mécanisme de prise de décision s'appuie en fait sur l'idée suivante. Si le temps d'interarrivée séparant les deux dernières visites de deux agents mobiles successifs à un nœud est suffisamment petit, alors cela pourrait s'interpréter par le fait qu'il y aurait un grand nombre d'agents présents dans le réseau. Dans ce cas, l'agent devrait s'autoéliminer en provoquant ainsi la diminution de la taille de la population d'agents dans le réseau. Si ce temps d'interarrivée est suffisamment grand alors cela pourrait s'interpréter par le fait que le réseau n'est pas surchargé. Dans ce cas, l'agent peut se cloner.

### B. Les limitations de l'approche

Amin et Mikler dans [1] se sont inspirés de la méthode de la colonie de fourmis. Hors, les fourmis peuvent être considérées comme des agents réactifs simples dotés des trois comportements suivants : cloner, s'autoéliminer, et migrer, comme cela a été défini dans [1]. Le choix d'un comportement par un agent est effectué d'une manière déterministe selon un algorithme réactif simple de type *si Condition  $C_i$  alors Actions  $A_i$* . La condition  $C_i$  est évaluée à l'aide des paramètres prédéterminés  $\lambda$ ,  $MIN$  et  $MAX$ , et du temps d'interarrivée  $\Delta t$  comme cela a été présenté ci-dessus.

Malgré que cette approche permette de réguler la taille de la population, elle présente cependant une limitation importante. Cette limitation concerne l'utilisation des paramètres de contrôles globaux et fixés a priori  $MIN$ ,  $MAX$  et  $\lambda$ . Par conséquent, l'algorithme de Amin et Mikler [1] n'est pas un algorithme de régulation adaptatif. En effet, les simulations de l'algorithme de Amin et Mikler [1] que nous avons effectuées ont montré que la taille de la population  $N_s$  autour de laquelle elle se stabilise dépend uniquement des trois paramètres  $MIN$ ,

$MAX$  et  $\lambda$  [2,3,16]. Plus précisément, ces paramètres fixes et prédéterminés non seulement constituent une information nécessaire aux agents pour sélectionner une et une seule action, mais de plus ce sont eux qui dictent la taille de la population à atteindre dans un réseau donné quelque soit la taille de la population initiale. Il faudrait alors être capable de déterminer les meilleurs paramètres  $MIN$ ,  $MAX$ , en fonction de la capacité du réseau. Cela peut être envisagé d'une manière empirique si le réseau est statique. Cette solution ne peut donc convenir lorsque le réseau est de nature dynamique (i.e., modifications de la topologie et du nombre des nœuds d'une manière aléatoire et imprévisible). En effet, dans le contexte d'un réseau à grande échelle dans lequel les ressources changent d'une manière aléatoire et imprévisible, il est nécessaire de concevoir des algorithmes de décision locaux au niveau des agents, sans recourir à une aucune information globale ou fixée a priori.

Une tentative pour améliorer cette approche consisterait à utiliser un mécanisme d'apprentissage pour permettre aux nœuds du réseau d'adapter d'une manière dynamique les trois paramètres en fonction des temps des interarrivées des agents reçus [2, 3, 16]. A titre d'exemple, si au niveau d'un nœud, le temps d'interarrivée est de plus en plus réduit, alors cela pourrait s'interpréter par le fait que le nombre d'agents présents dans le réseau est en augmentation. Dans ce cas, le nœud devrait ajuster le paramètre  $MIN$  pour renforcer l'élimination des agents reçus et provoquer ainsi la diminution de la taille de la population dans le réseau.

Cependant, comme l'approche s'appuie sur l'utilisation du temps d'interarrivée  $\Delta t$ , l'algorithme mis en oeuvre par Amin et Mikler [1] pour la régulation peut amener certains agents mobiles à prendre des décisions qui ne correspondraient pas à l'état réel. Plus précisément, considérons le nœud du réseau ayant le plus haut degré (i.e. le plus grand nombre de voisins). Ce nœud, par sa situation particulière au sein de la topologie du réseau, sera visité plus fréquemment que les autres nœuds du réseau. Par conséquent, l'intervalle  $\Delta t$  séparant deux visites successives pourrait être très réduit alors que le réseau n'est pas surchargé [2,3,16].

Il est important de noter aussi que le temps d'interarrivée des deux derniers agents successifs à un nœud, constitue une information qui ne rend pas compte de la taille de toute la population du réseau. Pourtant, les agents auraient besoin d'une information supplémentaire pour choisir les actions appropriées à l'état de la population. C'est pourquoi, Amin et Mikler [1] ont été amenés à utiliser les trois paramètres  $MIN$ ,  $MAX$  et  $\lambda$ , constants et fixés a priori. De plus, l'utilisation de l'apprentissage pour relaxer les paramètres  $MIN$  et  $MAX$ , et les rendre adaptatifs, au niveau des nœuds, ne nous a pas été possible comme cela a été démontré dans [2,3,16]. En effet, en se basant sur l'utilisation des temps d'interarrivées  $\Delta t$ , nous sommes confrontés à un environnement partiellement observable (i.e., un système distribué) dont l'information  $\Delta t$  fournie à l'agent est insuffisante. Il est nécessaire donc d'ajouter à un agent une information supplémentaire pour qu'il

puisse se comporter de façon optimale face à un environnement partiellement observable. Il s'agit d'une information constituée sur la base de l'expérience de l'agent.

Dans ce qui suit, une approche auto-adaptative inspirée du système immunitaire de l'homme qui nous permet de s'affranchir de ces paramètres prédéterminés et constants est présentée. Cette approche est basée sur l'utilisation d'un réseau idiotypique interne comme contrôleur pour l'agent qui peut permettre à ce dernier de mieux choisir son action en faisant appel à une information (i.e., un état interne) qui n'est pas immédiatement disponible dans son environnement [2,3,16]. Plus précisément, chaque agent de la population s'adapte, d'une manière autonome et sans recourir à aucune information globale, à l'état de l'environnement en choisissant le comportement le plus approprié. Ainsi, la régulation et la stabilisation de la taille de la population émergent d'une manière globale comme résultante des actions locales des agents autonomes distribués dans le réseau.

### III. UNE APPROCHE INSPIREE DU SYSTEME IMMUNITAIRE

#### A. Principe du système immunitaire utilisé

Le rôle du système immunitaire (SI) est de défendre le corps contre des agents pathogènes tels que les bactéries et les virus. Lorsque le SI détecte la présence d'antigènes (i.e., cellules spécifiques de "fragments" des pathogènes), une réponse immunitaire se met en place. Jerne [5] a établi que des cellules immunitaires, les lymphocytes B, communiquent entre elles en formant un réseau idiotypique constitué de chaînes de stimulation/suppression. Après une phase d'identification d'un antigène, les B-cells les plus appropriées produisent des anticorps spécifiques à cet antigène pour l'éliminer. Une B-cell est la plus appropriée lorsqu'elle possède la plus grande concentration, d'une part grâce à la stimulation des autres B-cell de son réseau et d'autre part grâce à son affinité avec l'antigène. Farmer, Packard et Perelson [4] ont modélisé les relations de stimulation et de suppression entre les B-cells, puis entre les antigènes et les B-cells par un système d'équations différentielles. Ce modèle mathématique a été largement repris dans plusieurs domaines, en particulier celui de l'intelligence artificielle, pour développer des solutions inspirées du système immunitaire [6]. Plus précisément, ce modèle de sélection inspiré du système immunitaire a été utilisé pour la première fois dans [14] par Watanabe et al., et dans [15] par Suzuki et al., pour résoudre deux problèmes : la collecte de déchets par un robot mobile autonome et la sélection de politiques d'affectation des ressources d'un serveur.

Par analogie avec le système immunitaire, les comportements sont les B-cells, et la chaîne de stimulation/suppression représente le réseau d'interactions entre ces comportements (i.e. le réseau idiotypique). Les concentrations d'anticorps générés par les B-cells sont représentées par des valeurs réelles associées aux comportements. L'antigène représente l'environnement perçu par l'agent. Une réponse immunitaire à un antigène est

représentée par le comportement ayant la concentration la plus élevée. Ce comportement émerge d'une manière indéterministe suite aux interactions entre les différents comportements et entre ces comportements et l'antigène.

Pour résoudre le problème de régulation de la population d'agents dans un réseau, nous utiliserons ce même mécanisme tout en s'inspirant également de l'approche de Amin et Mikler [1]

#### B. Principe de l'approche de régulation

L'analogie avec le système immunitaire est effectuée de la manière suivante. Un réseau idiotypique interne est incorporé à chaque agent mobile comme un contrôleur lui permettant d'effectuer le choix le plus approprié à l'état de son environnement. Plus précisément, les B-cells du réseau idiotypique représentent les trois comportements cloner, s'auto-éliminer et migrer. L'antigène représente le temps d'interarrivée  $\Delta t$ . Les trois comportements d'un agent sont reliés par une chaîne de stimulation/suppression comme cela a été établi dans le modèle original de Jerne [5]. Lorsque le réseau idiotypique est stimulé par un antigène, le comportement le plus approprié émergera suite aux interactions entre les B-cells d'une part et entre les B-cells et l'antigène d'autre part (voir la figure 1).

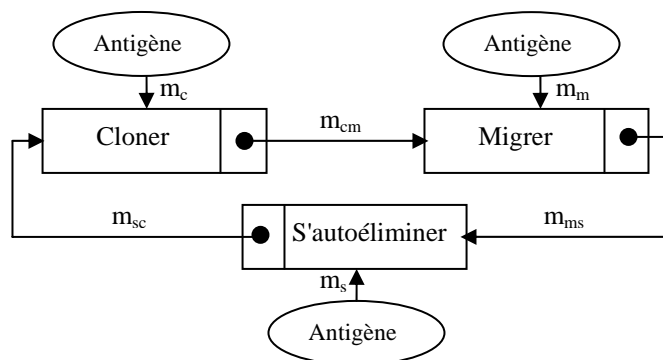


Fig. 1 – Les B-cell (i.e., les comportements) de l'agent sont reliés par les liens de stimulation et de suppression et sont stimulés en parallèle par l'antigène (i.e., les temps d'interarrivée).

Le graphe d'interconnexion entre les comportements est un cycle dans lequel, à titre d'exemple, le comportement cloner stimule le comportement migrer. Ce renforcement a pour objectif de permettre à un agent de pouvoir se cloner et de migrer au moins une fois avant d'être éliminé. De la même manière, le comportement cloner supprime le comportement s'autoéliminer. Plus précisément, lorsque l'antigène  $\Delta t$  est assez grand, il stimule le comportement se cloner en le renforçant positivement. Ce dernier renforce négativement le comportement s'autoéliminer. Le but est de permettre à l'agent de sélectionner le comportement se cloner.

Le mécanisme de décision du contrôleur est ainsi mis en oeuvre de la manière suivante. A chaque fois qu'un agent visite un nœud, son système immunitaire est stimulé par l'antigène  $\Delta t$ . Plus précisément, le comportement *cloner* est stimulé par le temps d'interarrivée  $\Delta t$  avec une affinité  $m_c$  égale à  $1 - \exp(-\Delta t)$ , le comportement *migrer* avec une

affinité  $m_m$  égale à  $\exp(-\Delta t)$  et le comportement *s'auto-éliminer* avec une affinité  $m_s$  égale à 1. Plus formellement, soient  $A_c$ ,  $A_m$  et  $A_s$  les concentrations associées respectivement aux comportements *cloner*, *migrer* et *s'auto-éliminer*. Leur variation est régie par les équations issues du modèle proposé par Farmer et al. [2,3,4] pour modéliser le réseau de Jerne de la manière suivante:

$$A_c(t+1) = A_c(t) + (m_{sc}a_s(t) - m_{cm}a_m(t) + m_c - K_c)a_c(t)$$

$$A_m(t+1) = A_m(t) + (m_{cm}a_c(t) - m_{ms}a_s(t) + m_m - K_m)a_m(t)$$

$$A_s(t+1) = A_s(t) + (m_{ms}a_m(t) - m_{sc}a_c(t) + m_s - K_s)a_s(t)$$

$a_c$ ,  $a_m$  et  $a_s$  sont déterminées par l'intermédiaire de l'équation  $a_i(t) = 1/(1 + \exp(0,5 - A_i(t)))$ . Ces valeurs sont initialisées à 0,01 dans notre expérimentation. Les constantes  $K_m$ ,  $K_c$  et  $K_s$  représentent le taux de suppression (i.e., mort naturelle) permettant de diminuer la concentration des comportements respectifs. Elles sont aussi initialisées à 0,001 dans notre expérimentation. Les variables  $m_c$ ,  $m_m$  et  $m_s$  représentent l'affinité de l'antigène avec les trois comportements correspondants, *cloner*, *migrer*, et *s'autosupprimer* comme cela a été décrit ci-dessus. Les affinités  $m_{cm}$ ,  $m_{ms}$ , et  $m_{sc}$  entre les trois comportements sont mises à 1.

Il est important de noter que, contrairement à l'approche de Amin et Mikler [1], le choix d'un comportement n'est pas effectué d'une manière déterministe selon un algorithme réactif simple de type *si Condition  $C_i$  alors Actions  $A_i$* , mais que les trois comportements de l'agent sont stimulés en parallèle et c'est le comportement qui a la plus grande concentration qui émergera et sera finalement sélectionné (i.e., la B-cell qui correspond le mieux à l'antigène). L'utilisation du réseau idiotypique d'une manière interne au sein de l'agent nous a permis ainsi de s'affranchir des valeurs de contrôle prédéterminées et fixée a priori  $MIN$ ,  $MAX$ , et  $\lambda$ . En d'autres termes, les agents mobiles sont dotés d'un mécanisme de décision local qui leur permet de sélectionner d'une manière autonome le meilleur comportement en fonction de l'état de leur environnement.

#### IV. SIMULATION

Nous avons comparé les deux approches à l'aide de simulations réalisées par le simulateur Starlogo<sup>1</sup>. C'est un environnement de modélisation programmable, basé sur le langage de programmation logo, destiné à modéliser et simuler le fonctionnement des systèmes naturels. Plus précisément, il est destiné à la simulation des systèmes d'agents situés dans un environnement. L'environnement est représenté par une grille dans laquelle chaque sommet est relié à ses huit voisins immédiats.

Les simulations de l'algorithme de Amin et Mikler [1] dans un réseau de 2500 nœuds, ont montré que la taille de la

population  $N_s$  autour de laquelle elle se stabilise dépend uniquement des trois paramètres  $MIN$ ,  $MAX$  et  $\lambda$ . A titre d'exemple, comme cela est illustré dans la figure 2, avec une population initiale  $N_0 = 1000$  et les paramètres  $MIN = 0.2$ ,  $MAX = 0.7$  et  $\lambda = 0.5$ , la taille de la population se stabilise autour de la valeur 2600. D'autres simulations effectuées avec les paramètres  $MIN = 0.3$ ,  $MAX = 0.6$  et  $\lambda = 0.5$  montre que la taille de la population se stabilise autour de la valeur 1800, comme cela est illustré dans la figure 3.

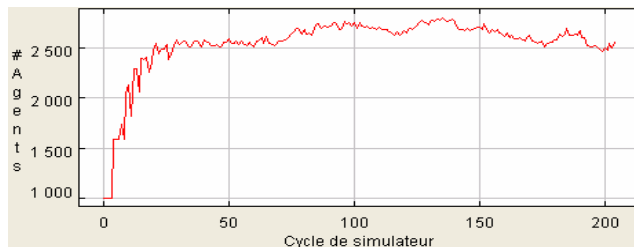


Fig. 2– Avec les paramètres  $MIN = 0.2$ ,  $MAX = 0.7$  et  $\lambda = 0.5$ , la taille de la population initiale  $N_0$  augmente dans un premier temps puis se stabilise ensuite autour d'une valeur  $N_s$  égale à 2600.

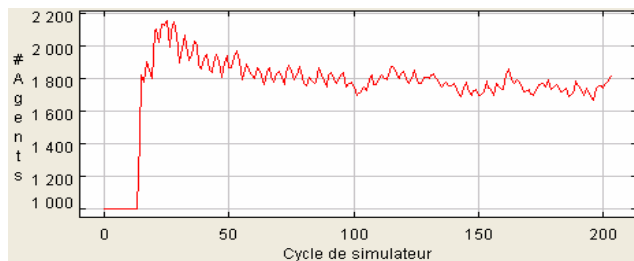


Fig. 3– Avec les paramètres  $MIN = 0.3$ ,  $MAX = 0.6$  et  $\lambda = 0.5$ , la taille de la population initiale  $N_0$  augmente dans un premier temps puis se stabilise ensuite autour d'une valeur  $N_s$  égale à 1800.

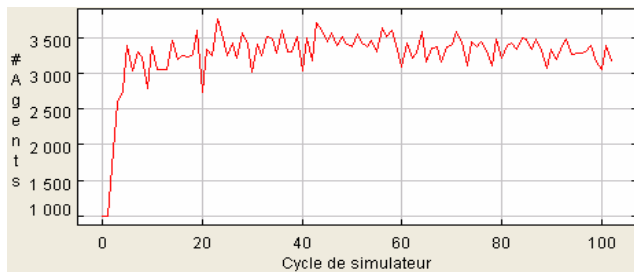


Fig. 4 – Régulation de la taille de la population d'agents dans un réseau de 2500 nœuds. La taille de la population initiale  $N_0$  augmente dans un premier temps puis se stabilise ensuite autour d'une valeur  $N_s$  égale à 3500.

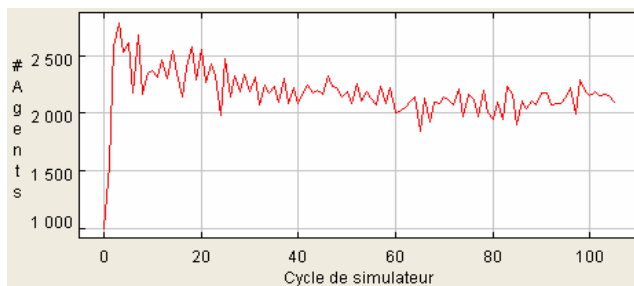


Fig. 5 – Régulation de la taille de la population d'agents dans un réseau de 1600 nœuds. La taille de la population initiale  $N_0$  augmente dans un premier temps puis se stabilise ensuite autour d'une valeur  $N_s$  égale à 2200.

<sup>1</sup> Starlogo simulator. <http://education.mit.edu/starlogo/>

Bien que cette approche inspirée du système immunitaire soit basée sur le même principe d'interprétation des temps d'interarrivées que l'approche inspirée de la méthode de la colonie de fourmis, sa mise en oeuvre est bien différente. En effet, contrairement à l'approche de Amin et Mikler [1], l'approche est mise en oeuvre en incorporant au contrôleur de chaque agent mobile un réseau idiotypique interne. Ce dernier lui permet de mieux choisir son action dans un environnement partiellement observable. Plus précisément, chaque agent de la population s'adapte, d'une manière autonome à l'état de l'environnement en choisissant le comportement le plus approprié sans recourir à des paramètres définis d'une manière globale et non adaptative. La régulation et la stabilisation de la taille de la population émergent d'une manière globale comme résultante des actions locales des agents autonomes distribués dans le réseau. De plus, les simulations effectuées ont montré que la taille limite  $N_s$  obtenue s'adapte en fonction de la capacité du réseau (i.e., le nombre de clones maximum pouvant être admis par le réseau) et que la taille de la population obtenue ne dépend pas de la taille de la population initiale  $N_0$ .

## V. CONCLUSION ET PERSPECTIVES

Dans cet article, nous avons présenté une approche distribuée et auto-adaptative pour la régulation d'une population d'agents dans un réseau. Cette approche est mise en oeuvre en incorporant au contrôleur de chaque agent mobile un réseau idiotypique interne inspiré du système immunitaire. Ce dernier lui permet de s'adapter d'une manière autonome à l'état de l'environnement en choisissant localement le comportement le plus approprié sans recourir à des paramètres prédéterminés non adaptatifs. Nous poursuivons les simulations en utilisant NS2.

## REFERENCE

[1] K. A. Amin and A.R. Mikler, "Dynamic Agent Population in Agent-based Distance Vector Routing", *Second International Workshop on Intelligent Systems Design and Applications*, Atlanta, USA, August 2002.

[2] M. Bakhouya, "Approche auto-adaptative à base d'agents mobiles et inspirée du système immunitaire de l'Homme pour la découverte de services dans les réseaux à grande échelle", *Thèse de doctorat*, UTBM, 2005.

[3] M. Bakhouya and J. Gaber, "Distributed auto-regulation approach of mobile agent population in a network", *Technical report*, SeT, Université de Technologies de Belfort Montbéliard (UTBM), pp. 1-14, December 2002.

[4] J. D. Farmer, N. H. Packard and A. S. Perelson, "The immune system, adaptation and machine learning", *Physica 22D*, pp 72-81, 1986.

[5] N. K. Jerne, "Towards a network theory of the immune system". *Ann. Immunol. (Inst.Pasteur)* 125C. 373, 1974.

[6] D. Dussupta, "Artificial Immune Systems and Their Applications", *Book*, Springer-Verlag, ISBN 3-540-64390-7, 306 pages, 1999.

[7] D. Hart, M. Tudoreanu, and E. Kraemer. "Mobile agents for monitoring distributed systems". *Proceedings of the Fifth International Conference on Autonomous Agents*, ACM Press, pages 232-233, 2001.

[8] V. Jamwal and S. Iyer. "Mobile agents for effective structuring of large-scale distributed applications". *Workshop on Software Engg and Mobility, International Conference on Software Engg (ICSE)*, Toronto, Canada, 2001.

[9] W. Theilman and K. Rothenmel, "Disseminating mobile agents for distributed information filtering". *ASAMA'99*, ISBN :0-7695-0340-3, pp. 152, 1999.

[10] S. Perret, "Agents mobiles pour l'accès nomade à l'information répartie dans les réseaux de grande envergure". *Thèse de doctorat*, Laboratoire LSR-IMAG, 1997.

[11] A. Sahai, "Conception et réalisation d'un gestionnaire mobile de réseaux fondé sur la technologie d'agents mobiles". *Thèse de doctorat*, Université de Rennes 1, 1999.

[12] M. Bakhouya, J. Gaber, "Observations on client-server and mobile agent based paradigms for resource allocation". *Workshop on Performance Modeling, Evaluation, and Optimization of Parallel and Distributed Systems PMEOPDS'02 (IPDPS 2002)*, Fort Lauderdale, Florida, April 15-19, 2002, pp.257-261.

[13] J.E. White, "Telescript technology: Mobile agents. Software Agents", J. Bradshaw, ed. AAAI Press/MIT Press, 1996.

[14] Y. Watanabe, A. Ishiguro, and Y. Uchikawa, "Decentralized behavior arbitration mechanism for autonomous mobile robot using immune system". *Books Artificial Immune Systems and Their Applications*, Springer-Verlag, p. 186-208, ISBN 3-540-64390-7, 1999.

[15] J. Suzuki and Y. Yamamoto, "A decentralized policy coordination facility in openwebserver". In Proc. of The SPA'00, 2000.

[16] M. Bakhouya and J. Gaber, "Adaptive approach for the regulation of a mobile agent population in a distributed network", In the 5th *International Symposium on Parallel and Distributed Computing (ISPD'06)*. Roumania. IEEE Press. Timisoara, Romania, July 6-9, 2006, pp.360-366.

[17] M. Wack, M. Bakhouya and J. Gaber, "Agent-based approach for Web crawling", In *International Transactions on Systems Science and Applications (ITSSA Journal)*. ISSN 1751-1461. Volume 2, N°2, pp.203-207, 2006.

[18] A. Broder, and A. Karlin, "Bounds on the cover time", *Journal of Theoretical Probability*, 2(1): 101-120, January 1989.

[19] J. Gaber, "New paradigms for ubiquitous and pervasive computing", *Research Report RR-09-00*, Université de Technologies de Belfort Montbéliard (UTBM), pp. 1-6, Septembre 2000.

[20] J. Gaber, "New paradigms for ubiquitous and pervasive applications", Proc. of the First Workshop on Software Engineering Challenges for Ubiquitous Computing, Lancaster, UK, June 2006.

[21] M. Bakhouya, and J. Gaber, "Approches de mise en oeuvre de l'ubiquité numérique", *Réseaux mobiles ad hoc et réseaux de capteurs sans fil (Traité IC2, série Réseaux et télécommunications)*, Ed. H. Labiod, Hermes, 02/2006, ISBN 2-7462-1292-7, pp.129-163.

[22] J. Gaber and M. Bakhouya, "Mobile agent-based approach for Resource discovery in Peer-to-Peer networks", In *Fifth International Workshop on Agents and Peer-to-Peer Computing (AP2PC)* at AAMAS. Mai 2006, pp.1-9.

[23] M. Bakhouya and J. Gaber, "Model-driven walks for resource Discovery in Peer-to-Peer", In *International Conference on Intelligent Agents, Web Technologies and Internet Commerce - (IAWTIC'2006, and CIMCA'2006)*, Sydney - Australia, December 2006. IEEE Press, to appear.

[24] J. Gaber and M. Bakhouya, "Approche Auto-adaptative inspirée du système immunitaire pour la mise en oeuvre de l'Ubiquité", In *3e Journées Francophones Mobilité et Ubiquité (UbiMob'06)*. CNAM, Paris. ACM Press. September 2006.

[25] J. Gaber and M. Bakhouya, "An Affinity-driven clustering approach for service discovery and composition for pervasive computing", In *Proceedings of IEEE International Conference on Pervasive Services ICPS'06*, 26-29 juin 2006, Lyon, France.

[26] D.B. Lange and M. Oshima, "Programming and deploying Java mobile agents with aglets", Addison-Wesley, ISBN 0-201-32582-9, 1998.

[27] M. Bakhouya and J. Gaber, "Self-organizing Approaches to Design Ubiquitous and Pervasive Applications", chapter in *Encyclopedia of Mobile Computing & Commerce (EMCC)*, Idea Group Pub, 2007, to appear.