



Application du système immunitaire artificiel ordinaire et amélioré pour la reconnaissance des caractères artificiels

Hiba Khelil^a, Abdelkader Benyettou^a

^aLaboratoire Signal Image Parole –SIMPA-, Université des Sciences et Technologie d'Oran, BP 1505 El M'naouer, Oran, Algérie

Résumé

Cet article traite la reconnaissance des caractères artificiels par les systèmes immunitaires artificiels. Nous définissons le système immunitaire biologique, sa simulation artificielle, ses algorithmes d'apprentissage et leur application à la reconnaissance des caractères artificiels. Nous présentons également le système immunitaire artificiel amélioré appliqué au même domaine. En conclusion, nous montrons l'impact de l'amélioration selon les deux critères : le nombre de cellules B générées et le taux de reconnaissance.

Mots clés: Système immunitaire naturel, Système immunitaire artificiel, Antigène, Anticorps, Cellules B mémoire, Reconnaissance des caractères artificiels.

1. Introduction

La reconnaissance des formes est une discipline vaste de l'intelligence artificielle. Elle couvre différents domaines tels que la reconnaissance des visages, les empreintes, la parole, l'écriture ... etc. Ces domaines font appel à plusieurs méthodes et approches telles que les réseaux de neurones, les algorithmes évolutionnaires, les algorithmes génétiques, ...etc. Le système immunitaire artificiel est une autre approche qui peut être appliquée à la reconnaissance des formes [1] [2] [3] [4] [5], mais aussi à la robotique [6], la détection des intrusions dans les réseaux informatiques [7], l'apprentissage machine [8] ... etc. Les fonctions du système immunitaire artificiel s'inspirent du système immunitaire naturel, notamment en ce qui concerne : les cellules responsables de la réponse ainsi que la réponse immunitaire. Ces facteurs peuvent contribuer à la construction d'un système proportionnellement complet destiné à la reconnaissance des formes.

Le présent travail est une application de l'approche immunitaire artificielle dans le cadre de la reconnaissance de l'écriture, et plus précisément les caractères artificiels. L'AIRS (Artificial Immune Recognition System) est une méthode inspirée du système immunitaire biologique orienté reconnaissance des formes proposée par A. Watkins en 2001 [9] à l'université de Mississipi. Une amélioration en a été faite en 2004 par A. Watkins, J. Timmis et L. Boggess [10] en optimisant le nombre de cellules B générées par apprentissage. Le SIA est une méthode

d'apprentissage distribué introduite par A. Watkins à l'université de Kent en 2005 [11].

Dans ce papier, nous commençons par une présentation du système immunitaire naturel et les types de réponse immunitaire. La deuxième partie est une représentation d'une simulation artificielle des systèmes immunitaires avec une explication des phases d'apprentissage. Un prototype d'apprentissage a été réalisé en se basant sur une base de données de caractères. Une étude comparative est montrée pour déduire les avantages de l'approche améliorée par rapport à l'approche d'origine.

2. Le système immunitaire naturel

Le système immunitaire biologique constitue une arme contre des intrus dans un corps donné. Pour ce faire, il existe plusieurs cellules qui contribuent à éliminer ces intrus nommé antigènes. Ces cellules participent pour ce qu'on appelle une 'réponse immunitaire biologique'. On distingue deux types principaux d'immunité naturelle : une innée et une acquise.

2.1. L'immunité innée

L'immunité innée est une immunité élémentaire qui est adaptée seulement à un certain nombre très réduit d'antigènes. On trouve ce type d'immunité chez les nouveaux nés qui ne sont pas encore vaccinés. Une immunité non adaptative pour longtemps peut conduire à

des infections et à la mort car le corps n'est pas encore bien protégé contre les antigènes de l'environnement [7].

2.2. L'immunité acquise

L'immunité acquise est une immunité à mémoire (réponse secondaire), qui se développe lors de l'apparition du même antigène dans le même système immunitaire pour la deuxième fois ou plus, et qui engendre le développement et la génération des cellules B mémoire pour ce type d'antigène déjà rencontré (mémorisé) dans le système. Cette réponse est plus rapide que celle innée [7]. Une réponse immunitaire engendre une augmentation de la température du corps, ce qui explique que les cellules B développées sont en train de lutter contre les antigènes introduits dans l'organe humain. La réponse immunitaire primaire est plus lente mais elle garde les informations du passage des antigènes dans le système. Il paraît intéressant de s'inspirer de ce phénomène de mémorisation pour une reconnaissance artificielle des formes.

3. Le système immunitaire artificiel

Le système immunitaire naturel est assez compliqué pour qu'une simulation artificielle soit réalisée d'une façon complète. Par contre, dans [9] l'auteur a réussi à simuler les fonctions les plus pertinentes dans un système immunitaire biologique pour que l'artificiel hérite le maximum des fonctionnalités naturelles dans le domaine de la reconnaissance des formes.

Les acteurs importants entrant dans un système immunitaire artificiel sont les antigènes, les anticorps et les cellules B mémoire. Dans la section qui suit nous présenterons l'algorithme d'apprentissage qui mettra en œuvre ces acteurs.

4. L'algorithme d'apprentissage du système immunitaire artificiel

Le présent algorithme est inspiré des travaux de A. Watkins [9] [12] [13] [14] nommé AIRS pour la reconnaissance des formes. Tout d'abord, l'antigène représente l'ensemble d'apprentissage dont il aura besoin pour le déroulement du programme, et le résultat d'apprentissage est l'ensemble d'anticorps (cellules B mémoire) qui seront utilisés pour la phase de test (les nouveaux antigènes introduits dans le système). L'apprentissage du système immunitaire artificiel se déroule en quatre étapes, chaque sortie est une introduction à l'étape suivante. Les étapes sont détaillées dans les points qui suivent:

4.1. Etape d'initialisation

Dans cette étape, il s'agit d'acquérir les données d'apprentissage (antigènes), ces données seront normalisées et tout vecteur caractéristique aura des valeurs de l'intervalle [0, 1]. Un seuil d'affinité est calculé à partir de cet ensemble d'antigènes, qui représente l'affinité moyenne entre tous les antigènes deux à deux. Le seuil d'affinité est calculé selon la formule (1) :

$$\text{seuil_d'affinité} = \frac{\sum_{i=1}^n \sum_{j=i+1}^n \text{affinité}(ag_i, ag_j)}{n(n-1)} \quad (1)$$

Avec : ag_i et ag_j deux antigènes et $\text{affinité}(a, b)$ retourne la distance euclidienne normalisée entre ag_i et ag_j .

Dans la phase suivante, une initialisation de l'ensemble des cellules mémoires (anticorps) et de la population des ARB¹ (Artificial Recognition Ball), est appliquée à partir de l'ensemble des antigènes par tirage aléatoire des exemples.

4.2. Etape d'identification des cellules B et génération des ARBs

Une fois que l'initialisation est achevée, cette étape aura lieu pour chaque nouvel antigène introduit. Une cellule mémoire est sélectionnée de l'ensemble entier de cellules B est nommée mc_{match} , cette dernière est tirée de telle sorte qu'elle ressemble² le plus à l'antigène en cours de traitement (la plus grande valeur de stimulation) selon la formule (2)

$$mc_{match} = \arg \max_{mc \in MC_{ag,c}} \text{stimulation}(ag, mc) \quad (2)$$

Une fois que la cellule mc_{match} est sélectionnée, elle sera utilisée pour générer les nouveaux ARBs (clones). Cet ensemble sera additionné à l'ensemble total des ARBs généré par l'ensemble des antigènes préalablement traités.

Le nombre de clones de la cellule mc_{match} est limité selon la formule (3):

$$\text{nombre_de_clones} = \text{hyper_clonal_rate} * \text{clonal_rate} * \text{stimulation}(mc_{match}, ag_{en_cours}) \quad (3)$$

¹ Le ARBs est un magasin de cellules B réduit, dont chaque ARB contient des informations sur la cellule B (anticorps, ressource, classe).

² La ressemblance est calculée par la formule de stimulation :

$$\text{stimulation}(ag, mc) = 1 - \text{affinité}(ag, mc)$$

Notons que chaque ARB généré par $m^{c_{match}}$ est muté selon l'algorithme de mutation décrit dans [9].

4.3. Compétition des ressources et développement des cellules mémoire candidates

Cette partie se base sur les résultats de la précédente, elle complète les informations des ARBs générés en calculant les ressources selon (4) pour chaque anticorps avec l'antigène en cours de traitement. Ces ressources sont mises à jour durant l'apprentissage, et chaque ARB n'ayant pas de ressources sera supprimé de l'ensemble des ARBs [9].

$$ressources = stimulation(ag, anticorps) * clonal_rate \quad (4)$$

Les parties 4.2 et 4.3 seront exécutés pour un antigène donné jusqu'à ce que la condition $s_i \geq Seuil_d'affinité$ soit vérifiée, avec

$$s_i = \frac{\sum_{j=1}^{|AB_i|} ab_j.stim}{|AB_i|}, ab_j \in AB_i \quad (5)$$

Une fois que la condition précédente n'est pas vérifiée on peut dire que nous avons pu avoir un ensemble d'ARBs optimal pour les introduire dans 4.4.

4.4. Introduction des cellules mémoire

Cette étape consiste à choisir parmi les ARBs la cellule candidate qui convient le plus à l'antigène ; en d'autres termes celle qui le reconnait mieux que $m^{c_{match}}$. La cellule candidate sera additionnée à l'ensemble des cellules mémoire seulement si elle retourne une stimulation plus élevée que la cellule de $m^{c_{match}}$ avec l'antigène en cours de traitement. Dans le cas contraire, la cellule $m^{c_{match}}$ sera retirée de cellules mémoire seulement si sa stimulation entre la cellule candidate avec $m^{c_{match}}$ ne dépasse pas le seuil d'affinité[9].

5. Les caractères artificiels

La base de données est un ensemble de caractères différents dont chacun est représenté dans différents fichiers et chaque fichier représente les coordonnées cartésiennes du caractère en cours. Voici l'exemple (fig. 1) d'un caractère qui est délimité par un ensemble de points du plan cartésien.

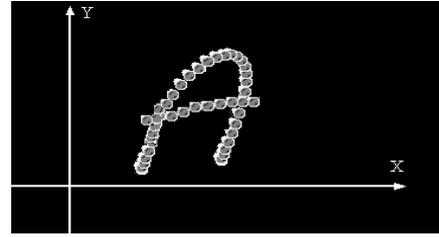


Fig.1 Exemple de caractère artificiel

Pour lancer l'apprentissage, nous avons choisi un corpus d'apprentissage de trois caractères (A, C et D) dont chacun contient 100 exemples. L'ensemble de test est constitué de 500 exemples pour chaque caractère. La table 1 montre la distribution des exemples.

Table 1

Nombre d'exemples du corpus d'apprentissage et de test

	Train	Test
A	100	500
C	100	500
D	100	500

6. Mise en œuvre

Pour la mise en œuvre de notre application, nous avons pris le corpus d'apprentissage de trois caractères donné dans la table 1, dont il représente les antigènes, et le but est d'atteindre en sortie un ensemble d'anticorps qui seront utilisés pour la phase de test.

Selon la table 1, on remarque bien que l'ensemble d'apprentissage est plus petit que l'ensemble de test, cette particularité caractérise ce type de système par rapport aux autres systèmes de la reconnaissance des formes. Cette particularité est tirée du principe biologique de vaccination, qui est l'introduction de petite quantité d'antigènes.

L'apprentissage nécessite de poser certains paramètres d'apprentissage, ces paramètres seront représentés dans la table 2 :

Table 2

Paramètres d'apprentissage

Paramètres	Signification	Plage de valeurs	Valeurs
hyper_clonal_rate	Taux maximal de clonage	Entier	25
clonal_rate	Taux moyen de clonage	Entier	15
mutate_rate	Probabilité de mutation	Réel $\in [0, 1]$	0.0001

6.1. Résultats et discussions

Le logiciel d'apprentissage est établi à l'aide du compilateur C++ sous l'environnement Linux Mandriva, déroulé dans une machine Pentium 4 doté d'un processeur de fréquence 2.4 GHz et d'une RAM de 512 Mo. Prenant en compte des paramètres indiqués dans la table 2, l'apprentissage est déroulé pour 50 itérations sur tout l'ensemble d'apprentissage. Les cellules B générés sont indiquées dans la table 3 et l'évolution de ces cellules par itération est montrée dans la figure 2.

Table 3

Cellules B générées après l'apprentissage

Cellules B générées au bout de la 50ème itération	
A	446
C	121
D	330

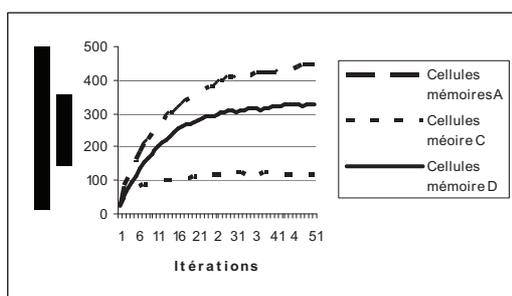


Fig. 2 Evolution des cellules B

Nous remarquons que les cellules B du caractère 'A' sont les plus générées par rapport à 'C' et 'D', cette différence est dû à l'importante variabilité des formes de 'A' par rapport à la variabilité des formes 'C' et 'D'.

6.2. Classification

Dans cette phase de test, nous allons utiliser les cellules B générées pour pouvoir classifier les nouveaux antigènes. Le principe est de pouvoir affecter un échantillon test

(antigène) à une classe de telle sorte qu'il se rapproche le plus des anticorps de cette classe en terme de distance. La méthode KMeans offre ce principe ; on calcule pour chaque classe de cellules B un centroïde qui sera utilisé pour calculer l'affinité avec l'antigène à classer : la classe qui donne la plus faible affinité sera celle sélectionnée pour cet antigène.

6.3. Résultats de classification

Se basant sur le principe de classification expliqué auparavant, les résultats de classification obtenus sont satisfaisants, la table 4 est une récapitulation des résultats de cette expérience.

Table 4

Résultats de classification

	Cellules B	Test	%Test	Apprentissage	%Apprentissage
A	446	500	100	100	100
C	121	500	66.73	100	67
D	330	500	100	100	100
			88.91		89

Dans la partie suivante nous allons appliquer l'amélioration apportée par A. Watkins, J. Timmis et L. Boggess [10] sur l'algorithme d'apprentissage des systèmes immunitaires artificiels.

7. Le système immunitaire amélioré

Cet algorithme est presque identique au précédent ; le changement opéré a été au niveau de la mutation et la compétition des ressources. En effet, dans la mutation, les auteurs ont introduit la notion de stimulation entre l'anticorps et l'antigène en cours de traitement. Il en est de même pour la compétition des ressources qui prend en compte seulement les anticorps de même classe que l'antigène [10] [14].

L'apprentissage s'est déroulé dans les mêmes conditions logicielles et matérielles que celle de l'expérience précédente, il en est de même pour les paramètres d'apprentissage qui sont les mêmes que ceux de la table 2. Les cellules B générées sont montrées dans la table 5, leur évolution est illustrée dans la figure 3.

Table 5

Cellules B générées après l'apprentissage

Cellules B générées au bout de la 50 ^{ème} itération	
A	356
C	62
D	77

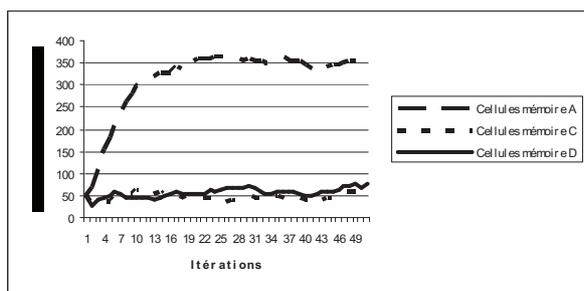


Fig 3 Evolution des cellules B

Table 6

Résultats de classification

	Cellules B	Test	%Test	Apprentissage	%Apprentissage
A	355	500	100	100	100
C	61	500	67.13	100	100
D	76	500	100	100	100
			89.04		100

Selon les résultats de classification obtenus par l'apprentissage du système immunitaire amélioré, on remarque bien que nous avons obtenu une amélioration de 0,134% sur la reconnaissance des caractères de test et de 11% sur l'ensemble d'apprentissage avec une réduction importante du nombre de cellules B générées dans la phase d'apprentissage. Cela veut dire que le système immunitaire amélioré est 13 fois plus précis que l'AIRS dans ces expériences appliquées.

8. Conclusion

Finalement, on peut dire que l'AIRS est un outil puissant dans la reconnaissance des formes. Nous avons vu aussi que le système immunitaire amélioré a un impact satisfaisant sur le temps d'exécution, le nombre de cellules B générées ainsi que sur le taux de reconnaissance.

L'intégration des AIRS dans d'autres méthodes d'optimisation pourrait apporter de meilleurs résultats en faisant le choix sur les meilleurs paramètres

d'apprentissage, en prenant l'exemple des algorithmes génétiques. Ce qui fera l'objet d'un prochain travail.

Références

- [1] Deneche A. (2006). *Approches bios inspirées pour la reconnaissance de formes*, Thèse de Magister de l'université Mentouri, Constantine, Algérie.
- [2] Deneche A., Meshoul S., Batouche M. (2005). Une approche hybride pour la reconnaissance de formes en utilisant un Système Immunitaire Artificiel, *Procédant des Journées d'informatique graphique*, Algérie
- [3] Emilie P. (2006). *Organisation du system immunitaire Felin*. Thèse doctorat vétérinaire, Ecole nationale de Lyon. Lyon
- [4] Goodman D., Boggess L., Watkins A. (2002). Artificial immune system classification of multiple class problems. *Journal Intelligent Engineering Systems Through Artificial Neural*.
- [5] Secker A., Freitas A., Timmis J. (2003). AISEC : an Artificial Immune System for E-mail Classification. *Journal IEEE Evolutionary Computation ECE'2003* pp 131-138.
- [6] Jun J. H., Lee D. W., Sim K. B., (1999). Realization of cooperative and Swarm Behavior in Distributed Autonomous Robotic Systems Using Artificial Immune System. *Procédant IEEE, conférence Man and Cybernetics*.
- [7] Kim J., Bently P. (2001), Towards an artificial immune system for network intrusion detection : an investigation of clonal selection with a negative selection operator, *Procédant de la conférence Congress on Evolutionary Computation*.
- [8] Timmis J., (2000), *Artificial Immune Systems: A novel data analysis technique inspired by the immune network theory*, PhD, Département d'informatique, université de Wales UK
- [9] Watkins A. (2001). AIRS : *A resource limited artificial immune classifier*. Thèse de l'université de Mississippi du département d'informatique. Mississippi.
- [10] Watkins A., Timmis J., Boggess L. (2004). Artificial immune recognition system (AIRS) : an immune inspired supervised learning algorithm. *Journal de l'academie Kluwer Genetic Programming and Evolvable Machines*. pp 291-317
- [11] Watkins A., (2005), *Exploiting immunological metaphors in the development of serial, parallel, and distributed learning algorithms*, PhD de l'université de Kent.
- [12] Watkins A., Boggess L. (2002a). A New Classifier Based on Resources Limited Artificial Immune Systems. *Procédant congress de Evolutionary Computation, IEEE World Congress on Computational Intelligence held in Honolulu*, pp 1546-1551. IEEE. USA.
- [13] Watkins A., Boggess L. (2002b). A Resource Limited Artificial Immune System. *Journal IEEE World Congress on Computational Intelligence held in Honolulu, USA* pp 926-931.
- [14] Watkins A., Timmis J. (2002). Artificial Immune Recognition System (AIRS): Revisions and Refinements. *1ère Conférence internationale du Système immunitaire artificiel ICARIS'2002*, pp 173-181, Université de Kent à Canterbury.