

Vie Artificielle : application à la résolution de problèmes complexes

par Daniel DUPONT* Annemarie KÖKÖSY** Philippe BIELA*** et Allal SAADANE****

La Vie Artificielle s'inspire du vivant pour recréer les principes qui le sous-tendent en introduisant de nouvelles formes d'entités, matérielles ou virtuelles, susceptibles d'évoluer dans des mondes synthétiques.

Les outils qui en découlent trouvent leurs applications dans de nombreux secteurs dont, par exemple, la robotique, la gestion de réseaux, le domaine des transports, l'infographie.

1. Objectif

La Vie Artificielle et ses principes associés (importance du collectif, notion d'émergence, intelligence collective ou auto-organisation) constituent une voie d'étude très riche, notamment pour la résolution de certains problèmes complexes ne possédant pas de solution analytique ou ayant une solution trop complexe pour être mise en œuvre facilement.

Cet article présente le domaine de la Vie Artificielle ainsi que trois techniques qui lui sont intimement liées :

- les automates cellulaires ;
- les algorithmes génétiques ;
- les systèmes multiagents.

Chacun de ces outils sera appliqué à un même problème complexe qui est la détermination d'un itinéraire vers la sortie d'un labyrinthe.

Le but n'est pas de comparer ces outils appliqués à la résolution de ce problème (chacun ayant ses propres qualités), mais d'illustrer ces techniques sur un même problème simple en apparence et bien connu.

2. La Vie Artificielle

La Vie Artificielle implique de nombreuses disciplines telles que la biologie, les sciences cognitives, l'informatique, les mathématiques, l'intelligence artificielle, etc., avec pour objectif de reproduire certains aspects du vivant. Dans les années 1940, Von Neumann (1903-1957) a initié l'étude du vivant au travers de modèles informatiques. En utilisant une configuration d'automate cellulaire, il met en évidence une des propriétés fondamentales du vivant : l'autoreproduction. Ses travaux ont été développés par d'autres scientifiques qui se sont consacrés à l'étude et à la compréhension du vivant afin de s'en inspirer pour synthétiser des systèmes artificiels.

Dans les années 1980, Langton porte un nouveau regard sur ce domaine : « La Vie Artificielle ne consiste pas seulement à étudier la vie existante, mais également à étudier la possibilité de synthétiser une nouvelle forme de vie au sein d'ordinateurs ou d'autres moyens artificiels » [1] [2]. La Vie Artificielle s'intéresse désormais aux bases formelles de la vie et l'envisage telle qu'elle pourrait être sans se limiter à ce qu'elle est.

Ainsi la Vie Artificielle repose sur l'abstraction des propriétés caractéristiques du vivant, puis sur leur implantation sur des systèmes artificiels [3]. Les systèmes informatiques représentent, bien entendu, un support de premier plan grâce à leur capacité de reproduire des environnements synthétiques imaginés par les concepteurs de programmes.

La démarche utilisée par la Vie Artificielle repose plus sur une synthèse fonctionnelle de la vie que sur son analyse fonctionnelle. Le vivant est vu comme un processus dynamique qui peut être implanté en suivant ses lois d'évolution.

Pour ce faire, deux étapes sont nécessaires :

- premièrement, identifier les principes qui le régissent ;
- deuxièmement, les implanter sur un système afin de les étudier et de les tester [2].

Ainsi le but de la Vie Artificielle est de comprendre le vivant et de recréer les principes qui le sous-tendent. Dans ce sens, on peut distinguer trois approches [4] :

- celle du modélisateur qui développe des modèles afin de faire progresser la connaissance du vivant ;
- celle de l'ingénieur qui conçoit des systèmes les utilisant comme des outils performants et efficaces ;
- celle de l'artiste qui utilise les capacités de l'ordinateur pour synthétiser des œuvres.

* Enseignant-chercheur à l'École supérieure des techniques industrielles et des textiles (ESTIT).

** Enseignant-chercheur à l'Institut supérieur d'électronique du Nord (ISEN).

*** Enseignant-chercheur à l'École des hautes études industrielles (HEI).

**** Enseignant-chercheur à l'Institut catholique d'arts et métiers (ICAM).

Membres d'ERASM : Equipe de recherche en automatique des systèmes et microsystèmes (Lille).

Dans cet article, on s'intéressera essentiellement à la deuxième approche en utilisant des techniques issues de la Vie Artificielle pour la résolution d'un problème complexe : **la détermination de l'itinéraire vers la sortie au sein d'un labyrinthe.**

2.1 Fondements de la Vie Artificielle

La Vie Artificielle est basée sur différents principes dont l'importance du collectif, la notion d'émergence et l'intelligence collective ou auto-organisation.

■ Importance du collectif

Même si les méthodes analytiques ont fait leurs preuves dans beaucoup de secteurs, il existe des problèmes dont la description se prête mal à une formalisation mathématique. Il n'est pas toujours possible de définir un ensemble de lois et de déterminer les variables pertinentes qui décrivent parfaitement le phénomène étudié.

La Vie Artificielle, contrairement aux méthodes analytiques, procède par synthèse suivant une démarche ascendante. Les modèles ou simulations cherchent alors à produire, à partir d'un ensemble d'éléments interagissant les uns avec les autres et avec leur environnement, des structures de plus haut niveau [3]. L'approche ascendante part de simulations et d'observations physiques et tente de synthétiser des comportements de plus en plus complexes. De fait, si l'on veut simuler ces comportements, il est nécessaire de mettre en relation les entités qui caractérisent le phénomène. Il faut donc concevoir une collectivité d'éléments qui interagissent. Chacun agit localement, de manière autonome et parallèlement. C'est dans cette perspective de collection d'éléments mis en relation que la structure complexe se définit.

■ Notion d'émergence

Dans la structure décrite précédemment, il n'y a pas de centralisation de fonction, de propriété ou de pouvoir. Chaque élément contribue au fonctionnement global sur la base d'actions locales et généralement simples. Les propriétés globales, qui nous apparaissent complexes, émergent de l'ensemble de ces actions élémentaires. Il n'y a pas de volonté de la part des différents acteurs à définir ce comportement complexe et global ; c'est l'interprétation de l'observateur qui leur attribue cette complexité.

■ Intelligence collective

Lorsque l'on s'intéresse aux comportements des sociétés d'insectes dits sociaux tels que les fourmis, les termites ou les abeilles, on est surpris de leur niveau d'organisation. Toutefois, une vision anthropomorphique a tendance à supposer qu'il existe une structure centralisée d'organisation : la reine ou un autre élément gère la société. Il n'en est rien. Les insectes ne possèdent pas un système cérébral suffisamment développé pour cela.

L'intelligence collective émerge, notamment, des relations entre les individus. Elle est le résultat d'un processus d'auto-organisation au cours duquel l'environnement et la société se structurent mutuellement [5]. Chaque entité répond aux stimulus grâce à des

règles simples. Cette réponse entraîne une modification de l'environnement (par exemple, par dépôts de traces chimiques : les phéromones) qui génère de nouveaux stimulus entraînant des actions locales qui modifieront en conséquence l'environnement. Il n'y a pas, dans ce processus, d'intelligence individuelle ou collective mais plutôt des actions « réflexes ». Toutefois l'ensemble de ces comportements donne l'apparence d'un haut niveau d'organisation : c'est le phénomène d'auto-organisation ou d'intelligence collective.

Cette présentation succincte de la Vie Artificielle et des principes associés, montre la richesse de cette voie d'étude, notamment pour la résolution de certains problèmes complexes ne possédant pas de solution analytique ou trop complexe pour être mise en œuvre facilement.

2.2 Outils issus de la Vie Artificielle

Le domaine de la Vie Artificielle s'étend autour de différents thèmes dont les principaux sont [6] :

- l'analyse de la dynamique des phénomènes complexes ;
- l'évolution des populations ;
- l'étude des phénomènes collectifs à partir de l'interaction d'un ensemble d'entités ;
- la réalisation d'animats, c'est-à-dire de créatures autonomes capables d'agir et de survivre dans un environnement non entièrement spécifié.

Le premier thème peut être abordé par l'utilisation des automates cellulaires (AC), le deuxième par l'utilisation des algorithmes génétiques (AG) et les deux autres par l'utilisation des systèmes multiagents (SMA). Ces différentes techniques, issues de la Vie Artificielle, seront présentées par la suite et seront appliquées à la résolution du choix du parcours d'un labyrinthe. Ce problème servira d'illustration pour permettre une meilleure compréhension de ces outils.

3. Les automates cellulaires (AC)

Cet outil caractérise parfaitement les principes de la Vie Artificielle car, à partir d'éléments au fonctionnement simple, les cellules, on simule des comportements complexes.

Un **automate cellulaire** est un système à une ou plusieurs dimensions composé de cellules élémentaires qui ne peuvent se trouver, à un instant donné, que dans un état appartenant à un ensemble d'états possibles.

Ainsi le temps et l'espace sont représentés de manière discrète. À chaque instant, l'évolution synchrone de l'état de toutes les cellules est calculée en fonction de règles locales de transition faisant intervenir le voisinage de la cellule concernée.

Le comportement global est ainsi défini par l'ensemble des comportements de chaque cellule et

l'on peut voir émerger des propriétés particulières en fonction des caractéristiques de l'automate cellulaire.

3.1 Caractéristiques

■ Domaine

Un automate cellulaire est composé de cellules qui peuvent être de forme et de taille quelconques. L'ensemble de ces cellules constitue le domaine qui peut être à une, deux ou n dimensions. L'application en définit les caractéristiques, mais généralement la dimension est de 1 ou 2.

■ Voisinage

Chaque cellule est en relation avec un certain nombre de cellules voisines qui définissent le voisinage. Il en existe une grande variété, mais les plus généralement utilisés sont le voisinage de Von Neumann (4 voisines) et le voisinage de Moore (8 voisines) (cf. en marge). Il faut toutefois préciser que la définition du voisinage dépend fortement de l'application.

■ Ensemble des états

L'ensemble des états de l'automate cellulaire est défini en fonction de l'application. Cet ensemble est généralement fini. Chaque cellule se trouve, à chaque pas de temps, dans un état appartenant à l'ensemble des états de l'automate cellulaire.

■ Règles de transition

Les règles de transition définissent le caractère dynamique de l'automate cellulaire et en assure son évolution. À l'instant n , l'état de chaque cellule est calculé à partir de l'état de ses voisines et de son état à l'instant $n - 1$.

■ Conditions aux bords

Si le domaine utilisé est borné se pose le problème des conditions aux limites de l'espace. Une cellule se trouvant au bord de l'espace ne dispose pas forcément de toutes les voisines nécessaires aux règles de transition (cf. en marge).

Afin de pallier ce problème, il existe différentes stratégies dont les plus communes sont :

- les voisines n'existant pas sont créées et supposées dans un état particulier immuable ;
- on attribue à l'espace une configuration torique, c'est-à-dire que l'on fait rejoindre les bords extrêmes de l'espace (les conditions aux bords sont dites périodiques).

3.2 Trajectoires

Une fois qu'un automate cellulaire est complètement défini, un de ses intérêts réside en l'observation et l'étude des suites d'états du domaine, calculées à partir de différentes conditions initiales. Ces suites définissent les **trajectoires de l'automate cellulaire**.

Une trajectoire d'un automate est une application qui à un temps t fait correspondre un état du domaine, deux états successifs étant liés par la fonction de transition.

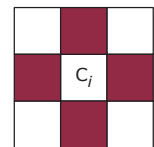
Une des propriétés les plus importantes de la trajectoire de l'automate est sa périodicité. Comme le nombre des états possibles pour chaque cellule, ainsi que le nombre de cellules, est fini, le nombre d'états

du domaine entier est, lui aussi, fini. Au bout d'un temps fini pouvant être important, on retombe donc forcément sur un état déjà rencontré. Pour une trajectoire donnée, on a deux notions de temps très importantes qui définissent de façon globale son comportement. La première, l'instant t_k , définit le transitoire de la trajectoire, tous les états de l'automate avant cet instant sont différents. La seconde, T , définit la période de la trajectoire, les états de l'automate à l'instant $T + t_k$ sont les mêmes que ceux de l'instant t_k .

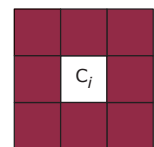
Exemple : soit un automate cellulaire à une dimension et de taille 3. Les cellules prennent leur état dans l'ensemble $\{0,1\}$. On suppose que l'évolution de l'automate est celle décrite dans le tableau 1. On peut remarquer qu'à l'instant $t + 4$ l'automate a le même état que celui de l'instant $t + 2$. Donc la longueur du transitoire est de 4. À l'instant $t + 6$ l'automate a le même état que celui de l'instant $t + 4$; la période T est égale à deux.

t	0	0	1
$t + 1$	0	1	0
$t + 2$	0	1	1
$t + 3$	1	0	0
$t + 4$	0	1	1
$t + 5$	1	0	0
$t + 6$	0	1	1
$t + 7$	1	0	0

Un **automate cellulaire** est caractérisé par cinq éléments : le domaine, le voisinage, l'ensemble des états, les règles de transitions et les conditions aux bords.



Voisinage de Von Neumann



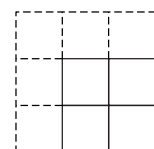
Voisinage de Moore

La longueur du transitoire est forcément finie mais on ne dispose pas d'informations a priori sur sa valeur. Pour certains automates cellulaires il est possible, par une approche analytique globale, de l'évaluer.

3.3 Le jeu de la vie

L'automate cellulaire le plus connu est celui inventé par Conway en 1970 nommé le jeu de la vie :

- le domaine est à deux dimensions et aussi grand que possible ;
- le voisinage est celui de Moore (8 voisines) ;
- une cellule est soit vivante (état 1), soit morte (état 0) ; c'est donc un automate cellulaire booléen ;
- les règles de transition sont :
 - une cellule vivante reste vivante si et seulement si elle possède 2 ou 3 voisines vivantes (condition de survie) ;
 - une cellule morte passe dans l'état vivant si et seulement si elle dispose de 3 voisines vivantes (condition de naissance) ;
 - dans les autres cas, la cellule meurt (étouffement ou isolement) ou reste vide ;
- généralement on lui associe une configuration telle que les conditions aux limites soient périodiques (configuration torique de l'espace).



Conditions aux bords

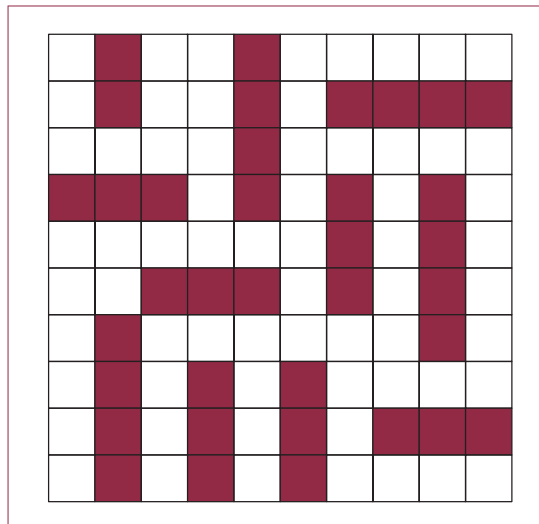
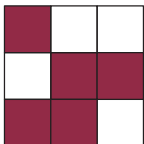


Figure 1 – Le labyrinthe

Cet automate cellulaire produit un comportement global d'une grande complexité et on peut voir émerger des structures particulières périodiques dont le **glisseur** qui est une forme qui se déplace diagonalement et un **canon à glisseur** qui produit des glisseurs. Cet automate cellulaire fait toujours l'objet de nombreuses études et de nouveaux résultats montrent la grande complexité des structures engendrées.



Glisseur

On retrouve dans cet exemple les caractéristiques des automates cellulaires et de la Vie Artificielle :

- toutes les cellules évoluent de manière synchrone et autonome, chacune en relation avec ses cellules voisines ;
- les règles d'évolution sont simples et n'agissent qu'à un niveau local ;
- le comportement global de l'automate cellulaire émerge de la structure et ne peut être expliqué par l'étude des comportements locaux.

3.4 Application des AC à l'exploration d'un labyrinthe

Appliquons le principe des automates cellulaires au problème de résolution de l'itinéraire dans un labyrinthe (figure 1). Dans ce schéma, les cases tramées représentent les murs et les cases blanches les lieux où l'on peut se déplacer.

Il est nécessaire de définir l'automate cellulaire par le domaine, le voisinage, l'ensemble des états, les règles de transition et les conditions aux bords.

■ Domaine

Comme nous pouvons le voir sur la figure 1, l'automate cellulaire est à deux dimensions, de taille 10 pour chaque dimension.

■ Voisinage

Comme on choisit de se déplacer dans le labyrinthe uniquement sur l'horizontale ou sur la verticale, le

voisinage le plus approprié est celui de Von Neumann (4 voisins).

■ Ensemble des états

Il y a 7 états possibles pour chaque cellule :

- *Indéterminé* : c'est l'état initial des cellules où l'on peut se déplacer ;
- *Impossible* : c'est l'état des cellules représentant un mur ;
- *Sortie* : c'est l'état de la cellule considérée comme la sortie du labyrinthe. Dans notre cas, c'est la cellule en bas à droite (case 10,10) ;
- *Nord, Est, Sud, Ouest* : ces états correspondent à une sorte de balisage qui définit dans quelle direction il faut se déplacer pour aller vers la sortie, direction nord, est, sud ou ouest.

■ Règles de transition :

- si, à l'instant $n - 1$, l'état de la cellule est *Indéterminé* et si une voisine est dans l'état *Sortie*, alors l'état à l'instant n est la direction à prendre pour atteindre la sortie ;
- si, à l'instant $n - 1$, l'état de la cellule est *Indéterminé* et si une voisine est dans l'état *Nord, Est, Sud* ou *Ouest*, alors l'état à l'instant n est la direction à prendre pour atteindre cette cellule voisine ;
- dans les autres cas, l'état de la cellule n'est pas modifié.

Le but final de ces règles est donc de baliser le labyrinthe. Ce balisage indique la direction à prendre pour atteindre la sortie, et ce, quelle que soit l'entrée choisie.

■ Conditions aux bords

Les « voisines » en dehors du domaine sont considérées comme étant dans l'état *Impossible*. On ne peut donc se déplacer en dehors du labyrinthe.

■ Évolution

La figure 2 montre l'évolution de l'automate cellulaire avec :

- a) la première itération : la case (9,10) est passée dans l'état *Est* indiquant ainsi qu'il faut se diriger vers la droite (les cases blanches sont dans l'état *Indéterminé* et les cases tramées dans l'état *Impossible*) ;
- b) évolution à la cinquième itération ;
- c) évolution à la septième itération ;
- d) évolution à huitième itération.

La figure 3 présente l'évolution à la dix-neuvième génération. Le résultat à la vingtième itération est le même que celui obtenu à la dix-neuvième itération. On peut conclure que le transitoire de la trajectoire de l'automate sous les conditions initiales données est de longueur dix-neuf. La périodicité T est égale à un, l'automate a atteint son état permanent. Lorsqu'il l'a atteint, l'état des cellules balise le chemin à suivre pour atteindre la sortie quelle que soit l'entrée du labyrinthe.

Supposons que l'entrée soit la case (1,1), la figure 4 montre le chemin à suivre pour atteindre la sortie. Comme nous pouvons le voir, le chemin quasi-optimal en termes de distance parcourue émerge en suivant l'état des cellules qui est calculé en fonction de l'environnement local à chaque cellule sans connaître l'état global du labyrinthe.

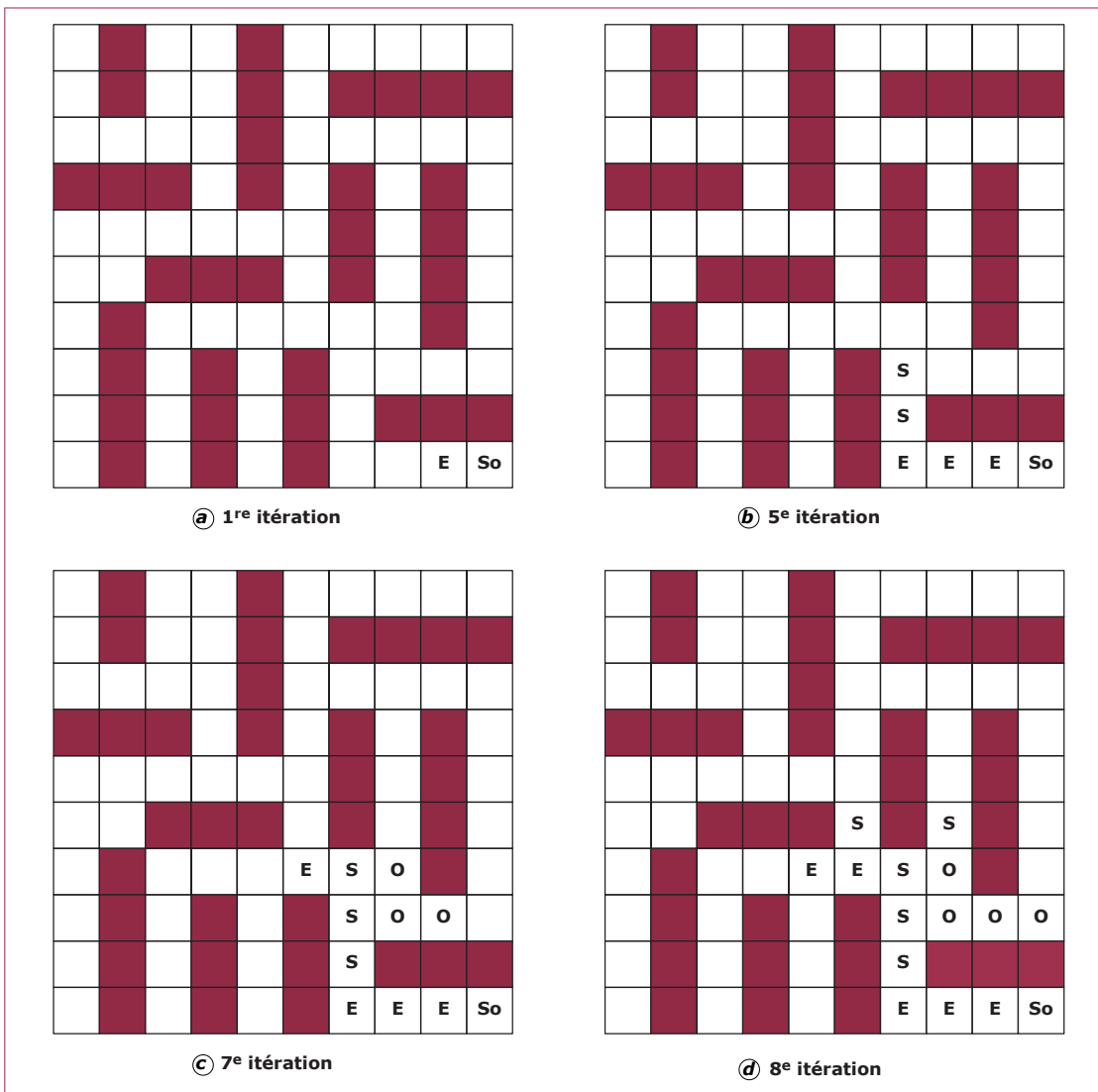


Figure 2 – Évolution de l’automate cellulaire

3.5 Conclusion

Cet exemple simple dans l’énoncé montre certaines capacités des automates cellulaires à résoudre un problème complexe.

Toutefois, il faut savoir que cet outil peut servir à la résolution de problèmes bien plus complexes en enrichissant les différentes caractéristiques ; par exemple, il est possible d’utiliser des règles de transition faisant intervenir des fonctions de probabilité construisant ainsi des automates cellulaires stochastiques.

Caractérisés par un très grand nombre de cellules et un faible transport d’information, les automates cellulaires sont capables de simuler une diversité de processus complexes comme on en rencontre dans

les systèmes dynamiques. Les applications des automates cellulaires sont nombreuses ; on les retrouve, par exemple, dans la simulation du comportement des gaz, du comportement des populations et du trafic urbain, de la propagation des feux de forêt, de la dynamique du tas de sable...

Si, dans notre exemple, on pouvait prédire que le chemin émergerait, bien souvent l’ensemble des interactions locales conduit à des comportements globaux difficilement prévisibles. Dans ces cas, la simulation au moyen de systèmes informatiques représente un atout indéniable. Le jeu de la vie en est un exemple où, en prenant une initialisation aléatoire du domaine, on peut voir émerger des formes particulières telles que le glisseur, sans pour cela avoir pu prédire qu’elle apparaîtrait. Il en est de

S		E	S		S	O	O	O	O
S		E	S		S				
E	E	E	S		S	E	S	O	O
			S		S		S		S
E	E	E	E	E	S		S		S
E	N				S		S		S
N		E	E	E	E	S	O		S
N		N		N		S	O	O	O
N		N		N		S			
N		N		N		E	E	E	So

Figure 3 – 19^e itération

S		E	S		S	O	O	O	O
S		E	S		S				
E	E	E	S		S	E	S	O	O
			S		S		S		S
E	E	E	E	E	S		S		S
E	N				S		S		S
N		E	E	E	E	S	O		S
N		N		N		S	O	O	O
N		N		N		S			
N		N		N		E	E	E	So

Figure 4 – Chemin à suivre si l'entrée est la case (1,1)

même pour une autre forme : le canon à glisseur qui génère, toutes les 30 itérations, un glisseur.

Von Neumann s'est intéressé à la notion d'autoreproduction (capacité de produire des copies de soi-même) en développant une machine appelée « Constructeur Universel » utilisant 29 états par cellule et le voisinage qui porte son nom. Cette machine donne une configuration extrêmement complexe. Langton proposa un automate cellulaire plus simple autoreproducteur qui utilise 8 états par cellules : les boucles de Langton.

Certains auteurs, dont Wolfram [7] et Langton, ont cherché à classifier les automates cellulaires en fonction de leur comportement dynamique. Il est intéressant de remarquer qu'ils concluent en disant que c'est à la frontière entre le chaos et l'ordre que les

comportements et les structures dynamiques les plus intéressants émergent.

4. Les algorithmes génétiques (AG)

Le père de la génétique est Gregor Mendel (1822-1884) qui, le premier, a su mettre en évidence la présence de caractères héréditaires dans le processus de reproduction chez les organismes vivants. Cette propriété d'hérédité est due aux gènes qui sont essentiels pour permettre la transmission d'informations d'une génération à l'autre. L'ensemble des gènes définit la carte génétique d'un individu. Ce vecteur informationnel permet aux individus d'être adaptés à leur environnement naturel mais également à l'espèce de s'adapter aux variations lentes de cet environnement. Ainsi, de génération en génération, ce sont des individus dont l'identité génétique révèle des caractères forts qui assureront la survie de l'espèce. La transmission génétique d'informations spécifiques permet aux organismes vivants de répondre en partie au problème de l'adéquation d'une espèce vis-à-vis des conditions de vie de son habitat naturel.

Les algorithmes génétiques sont inspirés de ce processus naturel de la transmission de l'identité des organismes vivants pour la résolution de problèmes spécifiques.

4.1 Contexte d'application des AG

La transmission des informations génétiques permet à une espèce de répondre au problème de l'adaptation. De façon simplifiée, on peut suggérer l'idée que ce principe naturel permet aux espèces vivantes de s'adapter aux conditions de vie locales en cherchant à favoriser la reproduction de la forme vivante la mieux adaptée au milieu. De façon imagée, il s'agit de la solution développée par la nature face à un problème d'optimisation. La transposition des principes génétiques mis en œuvre par la nature dans des techniques logicielles a conduit au développement des algorithmes génétiques constituant un outil pour la résolution de problèmes d'optimisation. Le paragraphe suivant présente les principaux mécanismes appliqués dans les AG, en soulignant pour chacun d'eux l'analogie existante avec les processus naturels de l'évolution.

Les AG mettent en jeu un certain nombre d'individus ayant pour objet la représentation d'une génération à une époque de vie donnée. Chaque individu représente une solution pour le problème d'optimisation posé. La transcription codée d'une solution prend généralement la forme d'une suite binaire, ce qui rend son exploitation informatique relativement aisée. Il est à noter à ce niveau que le terme de solution n'implique pas forcément une bonne solution ; il peut s'agir d'une très mauvaise solution, notamment dans la phase initiale de l'algorithme. Dans ces conditions, il s'agit avant tout de pouvoir juger de la performance d'une solution vis-à-vis du problème posé. On utilise pour cela une fonction d'évaluation qui permet de quantifier numériquement l'adéquation de l'individu, porteur d'une solution, avec son environnement. Dans le contexte naturel des organismes vivants, cette fonction d'évaluation pourrait

se réduire, par exemple, à constater la survie ou la mort de l'individu.

Les individus d'une même génération sont soumis au mécanisme de la reproduction, cela afin de permettre l'échange des caractères propres à chaque individu et d'initier pour la génération suivante la naissance de nouveaux individus avec une meilleure capacité intrinsèque d'adaptation au milieu. Cette reproduction va se faire, là encore, en imitant le processus des espèces vivantes. Dans leur milieu naturel, ce sont les individus les mieux adaptés (les plus forts et en bonne santé) qui ont le plus de chance de pouvoir s'accoupler et d'assurer ainsi la continuité de l'espèce. Au niveau des AG, c'est la fonction d'évaluation qui initie le processus de sélection en vue du mécanisme de reproduction. La reproduction est basée sur un principe d'échange des propriétés entre deux individus, elle implique la naissance de deux nouveaux individus. L'échange des informations s'effectue concrètement sous la forme d'un échange de parties de code entre les deux individus, cette opération porte le nom de croisement. Il est remarquable de constater que, si seuls les mécanismes de sélection et de croisement sont impliqués au niveau du processus de reproduction, il peut s'ensuivre un effet de dégénérescence de la population. Au cours des générations successives, les individus finissent par ne reproduire qu'une seule forme de solution disponible dont la qualité, mesurée par la fonction d'évaluation, peut se révéler médiocre. Il est nécessaire, pour éviter ce phénomène de dégénérescence, de copier une autre caractéristique génétique des organismes vivants, celle de la mutation. L'opération de mutation consiste à modifier ponctuellement la valeur du code génétique de quelques individus à chaque génération. Cette opération permet de générer de nouveaux individus porteurs de nouvelles caractéristiques, bonnes ou mauvaises.

4.2 Transposition des mécanismes génétiques aux problèmes d'optimisation

Ce paragraphe décrit plus précisément la transposition des mécanismes génétiques impliqués au niveau de la résolution de problèmes d'optimisation. Une description ordonnée de ces mécanismes est présentée, c'est-à-dire en respectant l'ordre chronologique de leur implémentation au niveau des AG.

■ Problème du codage

Si l'on désire chercher la solution à un problème d'optimisation à l'aide des AG, il est nécessaire, avant toute chose, de pouvoir représenter sous une forme codée l'expression des solutions.

Exemple : si le problème consiste à trouver l'abscisse de l'extremum d'une fonction mathématique $f(x)$ dont la variable est une grandeur réelle, on pourra exprimer la solution sous une forme binaire.

Il n'est cependant pas toujours facile de trouver l'expression codée de la solution, par exemple comment coder l'expression d'une solution de transaction boursière bénéficiaire pour un échange de titres dans un contexte économique mouvementé !

Le codage le plus couramment utilisé est le codage binaire car l'application des opérateurs de mutation et de croisement est plus facile à mettre en œuvre. Toutefois, ce type de codage n'est pas toujours suffisant en fonction de l'application. C'est pourquoi d'autres types de codage ont vu le jour comme le codage des gènes sous forme de nombres réels, par exemple.

■ Génération de la population initiale

Une fois le codage établi, il s'agit de générer un nombre d'individus représentatifs. Le choix de la population initiale conditionne la rapidité de convergence de l'algorithme. Si l'on dispose d'informations sur la localisation a priori de la solution, il est légitime de générer la population initiale dans une zone proche de cette solution. Dans le cas contraire, la population peut être générée aléatoirement sur tout l'espace des solutions. Ce dernier cas est le plus couramment utilisé.

Chaque génération successive sera composée du même nombre d'individus qui représente la taille initiale de la population. Il est fixé par l'utilisateur, mais il est raisonnable cependant de prévoir un nombre d'individus d'autant plus grand que le code de chaque individu est grand.

■ Évaluation d'une solution

Pour initier le processus de sélection et de reproduction, il faut pouvoir évaluer le degré de qualité de chaque solution par une fonction d'évaluation (*fitness function*).

Exemple : dans le cas de la fonction $f(x)$, la fonction d'évaluation sera la fonction $f(x)$ elle-même. Le passage par un optimum de la fonction sera détecté en examinant sa valeur pour chaque individu x .

Dans le cadre des opérations boursières, on peut retenir, comme fonction d'évaluation, le bénéfice réalisé.

■ Sélection des individus

Afin d'assurer l'amélioration des solutions au fil des générations successives, il est nécessaire de sélectionner des individus révélant des qualités naturelles d'adaptation à leur environnement. Cette qualité est mesurée par la fonction d'évaluation qui permet de sélectionner, par une procédure stochastique, les candidats parmi les meilleurs individus de la génération en cours. Les individus issus du mécanisme de sélection sont ensuite soumis aux procédures de croisement et de mutation.

■ Croisement

Le mécanisme de base assurant la reproduction des individus pour établir la génération suivante est le croisement (*cross-over*).

Pour le codage binaire, ce mécanisme relativement simple est illustré figure 5. Le croisement implique le choix préalable de deux individus parmi les individus sélectionnés et le choix aléatoire d'un point de croisement dans la chaîne génétique de ces deux individus. Il s'ensuit un échange partiel de l'information génétique entre les deux chaînes, qui donne lieu à la naissance de deux nouveaux individus.

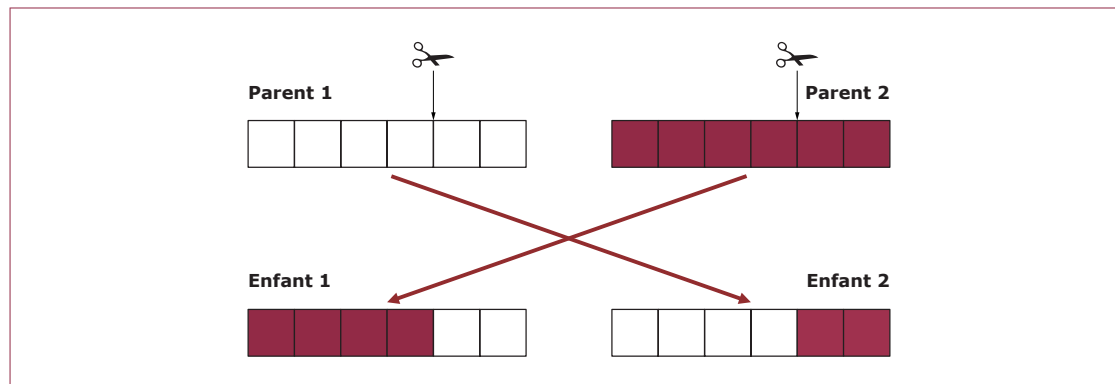


Figure 5 – Reproduction des individus au travers du mécanisme de croisement

Dans le cas du codage à valeur réelle, le croisement le plus souvent utilisé est un croisement barycentrique [8]. Un coefficient de pondération α est choisi aléatoirement, généralement entre 0 et 1, et le phénotype des enfants est construit par combinaison linéaire des phénotypes des parents. Deux gènes $P_1(i)$ et $P_2(i)$ sont sélectionnés aléatoirement dans chacun des parents à la même position i . Ils définissent deux nouveaux gènes $E_1(i)$ et $E_2(i)$ pour les enfants :

$$\begin{cases} E_1(i) = \alpha P_1(i) + (1 - \alpha) P_2(i) \\ E_2(i) = (1 - \alpha) P_1(i) + \alpha P_2(i) \end{cases}$$

Le principe de croisement est réitéré plusieurs fois en choisissant à chaque fois, au hasard, des couples d'individus. Le processus s'arrête lorsque le nombre de croisements choisi par l'utilisateur est atteint. Ce nombre est fonction du taux de croisement qui indique la proportion statistique d'individus susceptibles de participer au croisement au cours d'une génération. L'ordre de grandeur du taux de croisement est d'environ 25 % ; cette valeur est relativement commune à l'ensemble des problèmes traités par les AG.

■ **Mutation**

La mutation consiste à modifier de façon arbitraire la valeur codée d'un site de la carte génétique de l'individu. Le choix du site ainsi que de sa nouvelle valeur se font de façon aléatoire. Le taux de mutation doit être plus faible que le taux de croisement ; il peut être fixé à une valeur proche de 5 %.

De façon pratique, l'intérêt de cet opérateur est de permettre de balayer l'espace des solutions en ne restant pas bloqué dans une solution non optimale.

4.3 Application des AG à l'exploration d'un labyrinthe

Ce paragraphe met en œuvre les principes fondamentaux des algorithmes génétiques au travers d'un exemple d'application [9]. On expliquera successivement chacune des étapes essentielles des AG dans le contexte de la recherche d'un chemin vers la sortie pour un promeneur déambulant à l'intérieur d'un labyrinthe.

■ **Codage d'une solution**

Le choix du code pour la représentation d'une solution potentielle est déterminant pour assurer une bonne voie de recherche par les AG. Dans le cas du labyrinthe, il sera donc essentiel de pouvoir représenter de façon synthétique sa structure (entrée, sorties, murs) ainsi que les directions des déplacements susceptibles d'être empruntés par le promeneur en chaque point du labyrinthe (*Nord, Est, Sud ou Ouest*). Pour cela, on choisit de représenter l'architecture du labyrinthe par une matrice de cases de dimensions 10×10 et on assigne à chaque case un code qui spécifie s'il s'agit d'un mur ou de la direction prise par le promeneur. On obtient ainsi le code suivant :

- 0 pour une case inaccessible (*Mur*)
- 1 pour la direction *Nord*
- 2 pour la direction *Est*
- 3 pour la direction *Sud*
- 4 pour la direction *Ouest*

■ **Génération de la population initiale**

Dans le cadre du labyrinthe, chaque solution est codée sur 100 sites, on choisit arbitrairement de générer une population initiale de 24 individus, ce qui représente un nombre d'individus par génération a priori correct vis-à-vis de la longueur de code de chaque individu. Chaque site représente une case de la matrice du labyrinthe. Les sites représentant les murs sont codés à 0 et sont figés durant toute la procédure. Les autres sites sont codés à une valeur initiale choisie aléatoirement entre 1 et 4 et représentent la direction à prendre pour poursuivre le chemin.

■ **Fonction d'évaluation**

Le but de cette fonction est de mettre en évidence l'efficacité du parcours du labyrinthe ; elle est définie, après quelques essais, comme suit :

si, durant le chemin défini par la solution, on atteint la sortie [case (10,10)],

alors

la valeur de fitness est égale à 120 - longueur du chemin - nombre de demi-tours

sinon

si, le nombre de demi-tours est supérieur à la position atteinte

alors

la valeur de fitness = 0

sinon

la valeur de fitness est égale à la position atteinte - nombre de demi-tours.

Nota : la valeur 120 est prise, arbitrairement, afin d'avoir une fitness importante lorsque la sortie est atteinte et donc d'avoir une grande différence de fitness entre les solutions gagnantes (atteinte de la sortie) et les autres. Le fait de prendre 120 et non 100 (qui fonctionnerait également) permet d'avoir la meilleure fitness supérieure à 100, mais là encore ceci est arbitraire.

La position atteinte est égale à la somme des coordonnées de la case atteinte avant le blocage par un mur [on suppose que l'entrée est à la case (1,1) et la sortie à la case (10,10)].

Le fait de faire intervenir le nombre de demi-tours permet d'optimiser le chemin.

Face à la configuration du labyrinthe (figure 1), la longueur du chemin optimal est 18. De fait, la meilleure fitness atteignable est $120 - 18 = 102$ (le nombre de demi-tours étant nul).

■ Sélection

À l'issue de la phase d'évaluation, chaque individu est affecté d'une note proportionnelle à la qualité de la solution révélée par son code. La sélection va consister à extraire de la génération en cours un certain nombre d'individus en vue de préparer la génération suivante. Les individus sont sélectionnés suivant une procédure stochastique qui assure un choix prédominant pour les individus révélant les meilleurs scores. La taille de la population durant les générations successives doit rester identique à la taille initiale.

■ Croisement

Le croisement est appliqué sur une partie seulement des individus issus de la phase de sélection. La proportion des individus soumis au croisement est, dans le cas du labyrinthe, fixée à 25 %. Les individus sont choisis aléatoirement 2 par 2 et chaque paire de solutions parents va donner lieu à 2 nouvelles solutions enfants après échange des parties de code situées au-delà du point de croisement (figure 5). Le point de croisement est lui aussi choisi aléatoirement le long de la chaîne codée.

■ Mutation

La mutation consiste en une variation aléatoire de n'importe quel site de code appartenant aux solutions issues de la phase de sélection et ayant subies ou non une procédure de croisement. Le taux de mutation est également fixé a priori dès la phase initiale de l'algorithme. Dans le cas du labyrinthe, on choisit un taux de mutation égal à 2 %. Il ne doit pas être en effet trop élevé car il risquerait d'empêcher l'émergence progressive d'une solution en détruisant trop rapidement la structure génétique des individus adaptés à leur environnement.

■ Résultats

L'application des algorithmes génétiques suivant la procédure décrite auparavant a permis de mettre en évidence une solution, qui dans le cas présent, représente la solution optimale. Le nombre de générations successives nécessaires à l'établissement d'une solu-

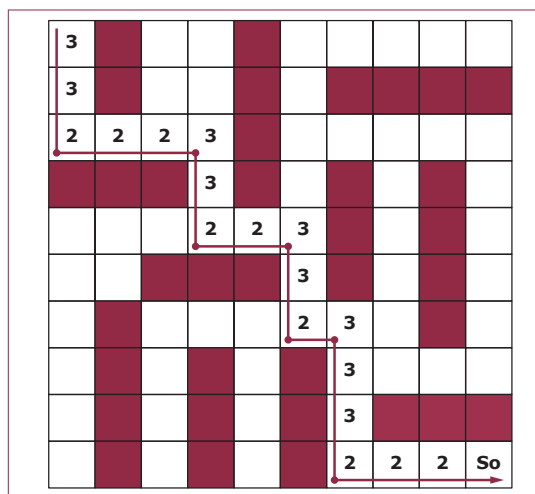


Figure 6 – Solution par les algorithmes génétiques

tion de chemin vers la sortie du labyrinthe a été généralement compris entre 1000 et 2000. La figure 6 montre la solution optimale obtenue à l'aide des AG.

4.4 Conclusion

Les algorithmes génétiques recherchent une « bonne » solution à un problème parmi un ensemble de solutions en tirant parti de l'historique de cette recherche afin de favoriser l'émergence des solutions les mieux adaptées. J.C. Heudin [4] caractérise les algorithmes génétiques de la façon suivante :

- les algorithmes génétiques travaillent sur la structure des solutions et non sur les solutions elles-mêmes ;
- les algorithmes génétiques explorent une population de solutions et non une unique solution ; on parle de parallélisme implicite de ces algorithmes ;
- les algorithmes génétiques utilisent une fonction d'évaluation objective, non une fonction dérivée ou nécessitant une connaissance auxiliaire ;
- les algorithmes génétiques utilisent des règles de transition probabilistes par l'utilisation d'opérateurs stochastiques et non des règles déterministes.

Ces différents points se rapprochent des caractéristiques que nous avons déjà introduites précédemment pour présenter la Vie Artificielle. En ce sens, les algorithmes génétiques représentent l'expression d'une forme de vie artificielle.

Toutefois, la recherche d'une solution par les AG ne garantit pas l'obtention d'une solution optimale. Une population initiale mal choisie (taille ou valeur), une convergence trop rapide vers un optimum local, etc., peuvent bloquer le processus de résolution.

Il n'existe pas de méthodes qui permettent de dire quelle est la taille optimale de la population, quel est le meilleur codage, quel taux de mutation et de croisement utiliser. Ces paramètres de fonctionnement dépendent de l'application et, bien souvent, ils sont fixés de façon empirique.

5. Les systèmes multiagents (SMA)

L'étude du comportement des insectes sociaux tels que les fourmis, les abeilles, les termites, révèle des modes de fonctionnement remarquables.

Chaque individu se comporte comme un acteur qui possède un script. Le script peut être défini comme l'ensemble des procédures de fonctionnement que l'acteur doit développer en réponse à des stimulus. Ces stimulus proviennent de l'environnement et des autres acteurs. L'exécution d'une procédure de fonctionnement aura pour effet de modifier l'environnement et donc de modifier ou de générer de nouveaux stimulus qui vont déclencher de nouvelles actions.

Cependant, il faut insister sur le fait qu'il n'y a pas de metteur en scène, il n'y a pas de plans globaux de fonctionnement. L'organisation de la société émerge de l'ensemble des interactions locales entre les différents acteurs. La construction des termitières en est un bon exemple car aucun individu n'en possède le plan de construction. Les individus répondent simplement à des stimulus à base de marquage chimique par les phéromones.

L'efficacité et la robustesse de telles organisations sont issues de ces interactions entre les individus eux-mêmes et leur environnement. On peut alors parler de fonctionnalités réparties.

Ces considérations ont amené les chercheurs à étudier la manière de formaliser les ingrédients fondamentaux de ce type d'organisation. L'objectif visé est de pouvoir disposer de nouveaux outils et de concepts adaptés à une approche « émergentiste » permettant, d'une part, de reproduire artificiellement le comportement de telles organisations et, d'autre part, d'utiliser ces outils pour la modélisation et la résolution de problèmes complexes où la démarche analytique est inopérante. Les domaines où cette approche semble parfaitement indiquée sont nombreux. On peut citer comme exemples :

- le transport : outils d'aide à la décision pour la gestion du trafic urbain... ;
- la robotique : réalisation de tâches complexes (exemple : exploration de Mars par un ensemble de robots mobiles effectuant des tâches élémentaires...) ;
- les NTIC (nouvelles technologies de l'information et de la communication) : moteurs de recherche pour un accès plus « intelligent » à l'information, optimisation des trafics dans les réseaux de communication...

L'essentiel de ces travaux de recherche s'articule autour de deux concepts fondamentaux : le concept d'agent et le concept de système multiagents [12].

5.1 Définitions

La définition de base la plus communément utilisée du terme agent est :

« entité autonome, réactive et communicante ».

Cette définition est souvent enrichie selon les domaines d'application (informatique, éducation, simulation, jeux, etc.) [10]. Concrètement, un agent peut avoir une réalité physique (robot, opérateur humain, etc.) ou logicielle dans le cadre d'un traitement informatique d'un problème complexe (approche distribuée).

Les systèmes multiagents (SMA) sont des architectures composées d'agents sur lesquelles l'expertise est distribuée. Ces systèmes se caractérisent par l'autonomie et l'indépendance des agents où chacun a un rôle à jouer et possède des compétences qu'il peut mettre au service des autres, et ce, sans avoir de vision globale de l'objectif à atteindre. En résumé, on considère généralement que, dans un tel système, chaque agent :

- peut communiquer directement avec d'autres agents ;
- possède des ressources propres, des compétences et offre des services ;
- est capable de percevoir partiellement son environnement et dispose d'une représentation partielle de celui-ci ;
- a un comportement qui tend à satisfaire ses objectifs ;
- peut apprendre de nouvelles compétences.

Face à ces définitions, le lecteur peut croire que l'utilisation des systèmes multiagents est complexe. Toutefois, l'ensemble de ces propriétés est modulable en fonction du degré « d'intelligence » attribué à chaque agent. Deux écoles coexistent :

- l'**école cognitive** où les agents sont capables de réaliser des opérations complexes et possèdent un certain niveau « d'intelligence ». On parle d'agents à forte granularité (*coarse grain*). L'Intelligence Artificielle Distribuée fait largement appel à cette approche ;
- l'**école réactive** où les agents ont des capacités réduites ; leur fonctionnement est alors plus simple, par contre ils sont beaucoup plus nombreux. On parle d'agents à faible granularité (*fine grain*). Cette approche est la plus couramment utilisée dans le domaine de la Vie Artificielle.

Dans l'école cognitive, le formalisme est beaucoup plus important. Les agents possèdent leurs buts, leurs planifications ; ils sont dits **intentionnels**. Dans l'école réactive, la démarche est plus empirique ; les agents réagissent aux événements sans avoir « conscience » des buts à atteindre ; leur fonctionnement est basé sur des relations stimulus/actions. La notion d'**émergence** y est fortement inscrite.

Le tableau 2 présente une classification des propriétés des deux types de SMA [11]. Bien évidemment, il est possible de concevoir des systèmes où les deux approches coexistent ; on parle d'agents hybrides.

Tableau 2 – Agents cognitifs et agents réactifs

Systèmes d'agents cognitifs	Systèmes d'agents réactifs
Représentation explicite de l'environnement	Pas de représentation explicite
Peut tenir compte de son passé	Pas de mémoire de son historique
Agents complexes	Fonctionnement stimulus/réponse
Petit nombre d'agents	Grand nombre d'agents

Cette présentation a pour objectif de présenter des outils issus de la Vie Artificielle qui, dans le domaine des systèmes multiagents, utilisent davantage les agents réactifs. La bibliographie présente d'excellents ouvrages qui permettront aux lecteurs d'approfondir leurs connaissances sur ce domaine [4] [5] [6].

5.2 Application des SMA à l'exploration d'un labyrinthe

L'utilisation des systèmes multiagents peut servir à résoudre le problème du parcours d'un labyrinthe (figure 1).

On utilisera un système d'agents réactifs bien qu'il possède des propriétés d'agents cognitifs : on pourrait parler d'agents hybrides.

À chaque agent est associée une structure de donnée (tableau) qui mémorise le trajet emprunté. Chaque cellule de ce tableau comporte la position (coordonnées x et y) par laquelle est passée l'agent.

Lorsqu'un agent se trouve dans une position, la position suivante à prendre est choisie aléatoirement ; on exclut évidemment la possibilité d'aller vers un mur et on exclut la possibilité de faire demi-tour sauf si l'agent se trouve dans une impasse [cas de la case (10,1) par exemple].

Les agents sont répartis par groupe de 4. Dans chaque groupe, on privilégie une direction à prendre (direction *Nord/Est*, *Est/Sud*, *Sud/Ouest*, *Ouest/Nord*). Cette prédominance à privilégier une direction peut être vue comme une notion de croyance dans le sens où l'agent « pense » que cette direction est meilleure. On a choisi d'utiliser 4 agents par groupe pour des considérations de performances informatiques.

Lorsqu'un agent atteint une impasse (obligé de revenir sur ses pas), il « marque » cette case comme étant interdite afin de communiquer à tous les agents, lui compris, qu'il est inutile d'aller dans cette direction. Cette communication est du type partage d'informations par l'intermédiaire d'un *blackboard* (figure 7).

Lorsqu'un agent atteint la sortie, il analyse son chemin, l'optimise et le communique à tous les autres agents par envoi de messages. L'agent « gagnant » optimise son chemin en supprimant les boucles (c'est-à-dire lorsqu'il est passé au moins deux fois par la même position) (figure 8).

Lorsque l'agent gagnant envoie sa trajectoire optimisée aux autres agents, ceux-ci optimisent leurs trajectoires suivant le même principe puis comparent

leurs trajectoires optimisées avec la trajectoire optimisée du gagnant et voient s'ils peuvent encore l'optimiser par rapport à leur historique. La figure 9 présente un exemple de trajectoire trouvée par le gagnant et une trajectoire trouvée par un autre agent. La comparaison aboutit à une meilleure trajectoire. Dans ce principe d'optimisation réciproque, il y a partage d'expérience, chacun communique à l'autre sa trajectoire et prend le meilleur de l'expérience de chacun. Les agents autres que le gagnant n'ont plus qu'à retrouver la position la plus proche commune à la meilleure trajectoire et à leur propre trajectoire puis atteindre la sortie.

Dans le cas où les agents doivent retourner au point de départ, il leur suffit de parcourir le chemin inverse. La solution trouvée n'est pas forcément le chemin optimal, mais en supposant qu'ils doivent continuellement faire des allers-retours, on peut imaginer d'avoir un ou plusieurs agents « récalcitrants » qui continueraient à explorer le labyrinthe et, dans le cas où ils trouvent une meilleure solution, la communiquent.

D'autres améliorations peuvent être imaginées, comme le fait que chaque agent analyse continuellement son propre chemin et, en fonction de son expérience, choisit une direction plutôt qu'une autre ; il y aurait alors phénomène d'autoapprentissage.

5.3 Conclusion

Comme on a pu le voir, les systèmes multiagents représentent un outil remarquable pour la résolution de problèmes complexes. De nombreux chercheurs continuent à les étudier pour cerner les potentiels qu'ils recèlent. Ces études sont orientées généralement selon les deux écoles dominantes, soit l'école réactive soit l'école cognitive. Néanmoins, l'approche mixte semble de plus en plus adoptée.

Les domaines d'application des SMA sont nombreux et variés. Ils sont utilisés dans le domaine de la robotique, du génie logiciel, de l'enseignement, de la gestion de réseaux, de système de surveillance, dans la gestion de trafic aérien ou urbain, etc.

6. Conclusion générale

Cet article a présenté différents outils issus de la Vie Artificielle pour la résolution du problème de parcours d'un labyrinthe. Le but n'était pas de les comparer, chacun ayant ses propres qualités, mais de les introduire et de les appliquer sur un exemple commun. Ce problème peut s'apparenter au problème de recherche de trajectoire dans un environ-

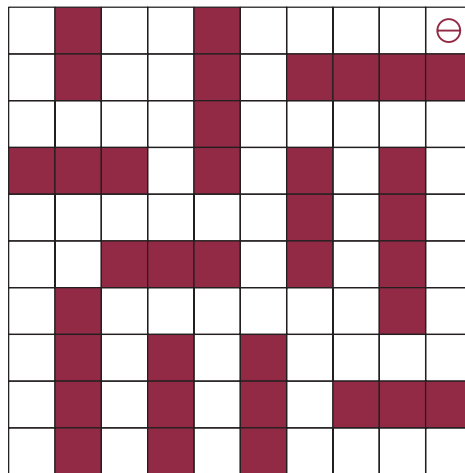
nement non complètement formalisé, peuplé d'obstacles, qui peut trouver des applications en robotique, en gestion de réseaux (informatique, de distribution...), dans le domaine des transports (par exemple gestion de trafic routier ou aérien), etc.

Il ne faut pas limiter la Vie Artificielle aux trois outils présentés ; ceux-ci nous ont servi dans un problème d'optimisation. La Vie Artificielle est un

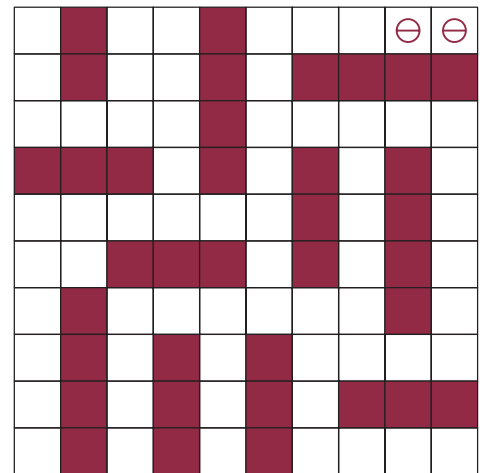
domaine beaucoup plus riche qui tente de créer de nouvelles formes d'entités vivantes.

Exemple : les L - systèmes permettent de simuler le développement des plantes et sont largement utilisés en réalité virtuelle.

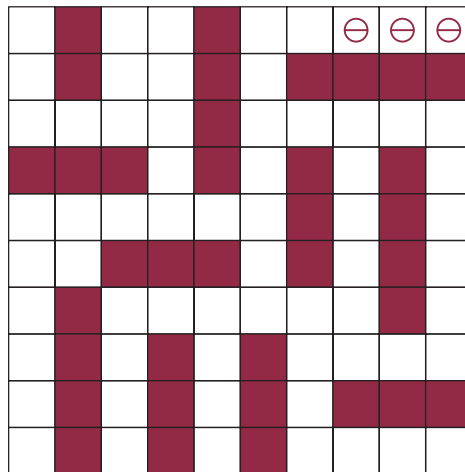
De même, des programmes permettent de générer des formes qui ressemblent étrangement à des êtres vivants nommés *Biomorphs*.



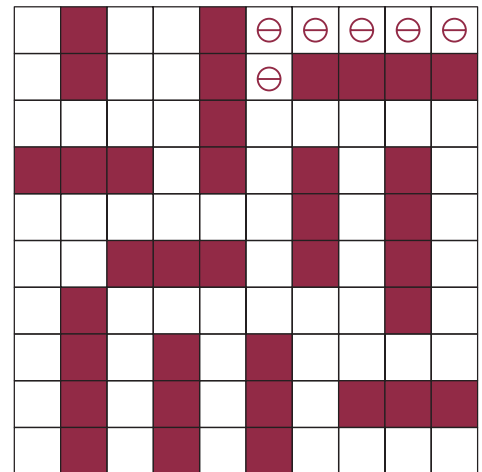
Ⓐ à la n^e génération, un agent se trouve dans la case (10,1) et constate que c'est une impasse



Ⓑ à la $n + 1$ génération, l'agent se trouve dans la case (9,1) et constate que c'est devenu une impasse



Ⓒ à la $n + 2$ génération, l'agent se trouve dans la case (7,1) et constate que c'est devenu une impasse



Ⓓ à la $n + 5$ génération, l'agent se trouve dans la case (6,2) et constate que c'est devenu une impasse

Figure 7 – Indication de présence d'une impasse

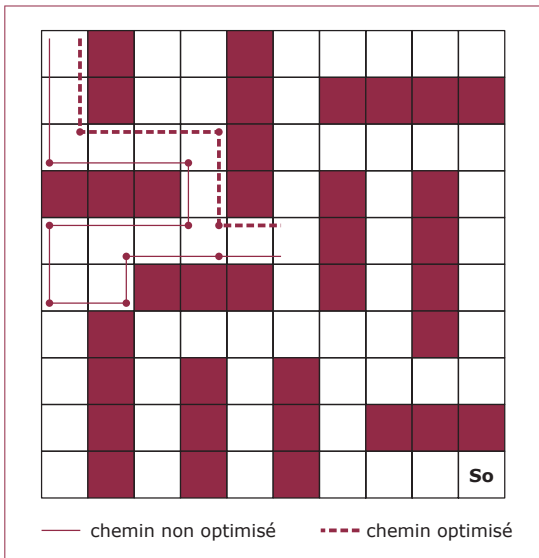


Figure 8 – Optimisation du chemin noir par le chemin vert

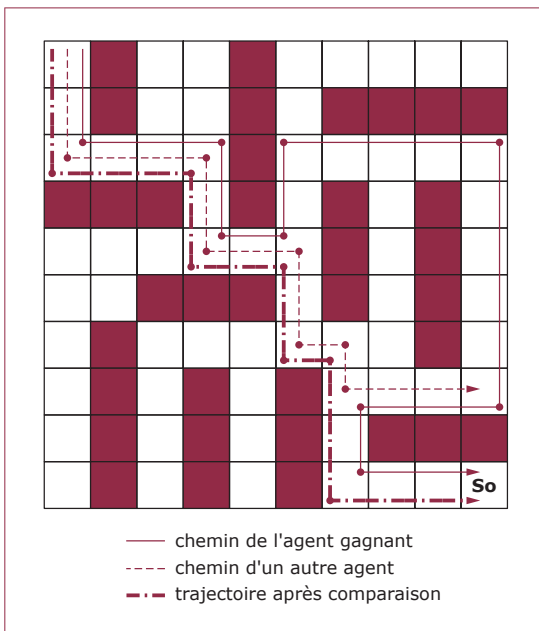


Figure 9 – Optimisation réciproque du chemin

Volontairement, on n'a pas parlé des réseaux neuromimétiques qui s'inspirent des structures neuronales car ils sont déjà bien connus et utilisés [13].

Bien évidemment, cet article n'a pas la prétention d'avoir tout dit sur les outils présentés et encore moins sur la Vie Artificielle. Ce domaine est en plein développement et promet de nombreuses perspecti-

ves. Le lecteur intéressé pourra se référer aux quelques ouvrages cités en bibliographie mais aussi consulter les nombreux sites web en utilisant par exemple les mots clés suivants.

Mots clés

Vie Artificielle, automate cellulaire, algorithme génétique, système multiagent, auto-organisation, émergence, intelligence collective, intelligence artificielle distribuée.

Keywords

Artificial Life, cellular automaton, genetic algorithm, multiagents system, self-organisation, emergence, collective intelligence, distributed artificial intelligence.

Contact :

erasm@fupl.asso.fr

Sites Internet

<http://www.vieartificielle.com>
<http://www.rennard.org/alife>
<http://www.cs.bgu.ac.il/~sipper/caslinks.html>

Références bibliographiques

- [1] LANGTON (C.). – *Artificial Life* (Vie Artificielle). Proceeding of the 1st Workshop on the Synthesis and Simulation of Living Systems, Addison-Wesley (1989).
- [2] LANGTON (C.), TAYLOR (C.), FARMER (J.D.) et RASMUSSEN (S.). – *Artificial Life II* (Vie Artificielle II). Proceeding of the 2nd Workshop on Artificial Life, Santa Fe, New Mexico, Addison-Wesley (1990).
- [3] BONABEAU (E.) et THERAULAZ (G.). – *Vie Artificielle et sciences du vivant*. 1^{er} Congrès biennal de l'association française des sciences et technologies de l'information et des systèmes, Versailles, p. 167-183 AFCET 93 (1993).
- [4] HEUDIN (J.C.). – *La Vie Artificielle*. Éd. Hermes, Paris, 261 p. (1994).
- [5] BONABEAU (E.) et THERAULAZ (G.). – *Intelligence Collective*. Éd. Hermes, Paris, 288 p. (1994).
- [6] FERBER (J.). – *Les systèmes multiagents. Vers une intelligence collective*. Éd. InterEditions, Paris, 522 p. (1995).
- [7] WOLFRAM (S.). – *Theory and applications of cellular automata*. World Scientific Publishing, 560 p. (1986).
- [8] ALLIOT (J.M.). – *Techniques d'optimisation stochastique appliquées à certains problèmes du trafic aérien*. Habilitation à diriger les recherches, Institut national polytechnique de Toulouse (INPT), Laboratoire d'informatique et mathématiques appliquées de l'ENSEEIH, 187 p. (1996).
- [9] DESSALLES (J.L.). – *L'ordinateur génétique*. Éd. Hermes Paris, 144 p. (1996).
- [10] ROUTIER (J.C.) et MATHIEU (P.). – *Une contribution du multiagent aux applications de travail coopératif*. Rubrique présentation, publications, TSI Télé-applications [<http://www.lifl.fr/SMAC/projets/magique/>].
- [11] LABIDI (S.) et LEJOUAD (W.). – *De l'Intelligence Artificielle Distribuée aux systèmes multiagents*. Rapport n° 2004, INRIA, Rocquencourt, 43 p. (1993).

Dans les Techniques de l'Ingénieur

- [12] MANDIAU (R.) et GRISLIN-LE STRUGEON (E.). – *Systèmes multiagents*. [S 7 216]. Traité Informatique industrielle (2002).
- [13] WEINFELD (M.). – *Réseaux de neurones*. [H 1 990]. Traité Informatique (1995).

Dans les Techniques de l'Ingénieur

Réseaux de neurones
H 1 990