
Approche évolutionniste de la recherche d'automates cellulaires universels

Emmanuel Sapin

Université de Bourgogne, 9 av. A. Savary, B.P. 47870, 21078 Dijon Cedex, France
emmanuelsapin@hotmail.com

RÉSUMÉ. Le fameux problème de la fréquence des automates cellulaires universels est liée à l'émergence de la calculabilité dans les systèmes complexes avec des interactions locales. Il a été posé une première fois par Wolfram : « How common are computational universality and undecidability in cellular automata ». Ce papier décrit des algorithmes évolutionnaires donnant des éléments de réponse à cette problématique. Nous présenterons un algorithme évolutionnaire découvrant des automates acceptant des planeurs puis un autre algorithme découvrant des canons à planeurs acceptés par ces automates. Un de ces automates est démontré universel et notre démonstration est généralisable aux autres automates découverts.

ABSTRACT. The famous problem of the frequency of universal cellular automata is related to emergence of computation in complex systems with simple local behaviour. It was asked by the first time by Wolfram : « How common are computational universality and undecidability in cellular automata ». This paper includes evolutionary algorithms providing elements of answer to this problem. An evolutionary algorithm that finds automata accepting glider is described. Then we elaborated another algorithm that finds glider gun accepting by these automata. One of these automata is shown universal and our demonstration can be generalized to other automata that we discovered.

MOTS-CLÉS : Algorithmes Évolutionnaires, Automates Cellulaires, Universalité, Planeurs, Isotropie, Canons à Planeurs, Jeu de la Vie.

KEYWORDS: Evolutionary Algorithms, Cellular Automata, Universality, Gliders, Isotropy, Glider Guns, Game of Life.

1. Introduction

Les automates cellulaires sont des systèmes discrets (Wolfram, 1983) dans lesquels une population de cellules évolue génération après génération, sur la base d'une règle de transition locale. Ils peuvent être utilisés pour simuler des formes de vie simplifiées (Gardner, 1970)(Gardner, 1971) ou des systèmes régis par des interactions locales. Ils sont notamment utilisés pour prévoir l'évolution des feux de forêt (Drossel *et al.*, 1992) ou pour générer des nombres aléatoires (Wolfram, 1986) mais aussi dans des domaines tels que la chimie (Epstein, 1991), la biologie (Ermentrout *et al.*, 1993) ou encore l'économie (Oda *et al.*, 1977).

La machine de Turing est une machine permettant le calcul de toutes fonctions calculables (Turing, 1948). Le propre d'un ordinateur, dans sa définition théorique, est sa capacité à simuler une machine de Turing ou une machine équivalente à une machine de Turing. Cette capacité, introduite par Turing, est une caractéristique fondamentale appelée universalité.

Nous allons nous intéresser au problème de la fréquence des automates cellulaires universels. Ce problème est liée à l'émergence de la calculabilité dans les systèmes complexes avec des interactions locales. Il a été posé une première fois par Wolfram dans (Wolfram, 1985) : « How common are computational universality and undecidability in cellular automata ». Un automate universel peut simuler une machine équivalente à une machine de Turing, il est donc capable de simuler n'importe quel autre automate ou d'effectuer des calculs non parallèles. La recherche d'universalité des automates est une question centrale depuis l'automate de Von Neumann et Burks qui avait la capacité d'autoréplication (Neumann *et al.*, 1966). Les cellules de cet automate pouvaient prendre 29 états et la fonction de transition d'un automate est d'autant plus simple que cet automate a peu d'état. Pour faire émerger l'universalité à partir d'une règle locale la plus simple possible Codd a trouvé un automate autoréplicateur ayant 8 états (Codd, 1968). D'autres recherches sur des automates universels à plus de deux états ont aussi été menée par Lindgrel (Lindgrel *et al.*, 1990) ou Morita (Morita *et al.*, 2002) tandis que la recherche d'universalité dans des automates dont la fonction de transition prend en compte un voisinage hexagonal ou de Von Neumann a été explorée par Margolus (Margolus, 1988) ou encore Banks (Banks, 1970).

Afin de conserver la simplicité de la règle de transition, nous nous limiterons aux automates à deux états, deux dimensions et dont la fonction de transition prend en compte les huit voisins d'une cellule pour déterminer l'état de cette cellule à la génération suivante. Dans cet espace d'automates, qui avait déjà été étudié par Wolfram (Wolfram *et al.*, 1985) et que nous appellerons *E*, très peu d'automates sont démontrés universels. Parmi les automates de *E*, le jeu de la vie, découvert par Conway en 1970 et popularisé par Gardner dans (Gardner, 1970), a été démontré universel par Conway de manière constructive (Berlekamp *et al.*, 1982). Cette démonstration utilise des planeurs et des canons à planeurs. Les planeurs sont des formes périodiques qui, si elles évoluent seules, se reproduisent à l'identique en se décalant dans l'espace. Les canons à planeurs produisent des suites de planeurs qui

permettent de transporter des informations et de réaliser, par collision, des portes logiques. Rendell (Rendell, 2002) donne une preuve explicite de l'universalité du jeu de la vie en montrant la simulation directe d'une machine à registres.

La fonction de transition définissant entièrement un automate de l'ensemble E étant de $\{0,1\}^9$ dans $\{0,1\}$, il se compose de 2^{512} automates. Cet ensemble étant très vaste, nous avons privilégié une approche évolutionniste pour la recherche d'automates universels de E . Nous avons conçu un algorithme évolutionnaire cherchant des automates de E acceptant des planeurs. Pour trouver des automates ayant des planeurs multidirectionnels, cet algorithme a ensuite été modifié pour rechercher des automates isotropes. Cet algorithme a permis de découvrir l'automate universel R décrit dans (Sapin *et al.*, 2004 b). La démonstration de son universalité s'appuie sur la présence dans cet automate d'un canon à planeurs. La découverte de nouveaux automates cellulaires acceptant des canons à planeurs semble donc être une voie prometteuse vers la découverte de nouveaux automates universels. Nous avons donc élaboré un algorithme évolutionnaire permettant de faire apparaître des canons à planeurs acceptés par des automates cellulaires.

Dans le paragraphe 2, nous allons décrire notre démarche de recherche. Les conventions de codage utilisées seront décrites dans le paragraphe 3. Le paragraphe 4 décrit un algorithme évolutionnaire recherchant des automates de E qui acceptent des planeurs et des formes périodiques. La recherche par algorithme évolutionnaire d'automates isotropes acceptant des planeurs est décrite dans le paragraphe 5. Le paragraphe 6 décrit la recherche par algorithme évolutionnaire de canons à planeurs parmi les automates découverts. Nous présentons enfin, dans le dernier paragraphe, une synthèse des résultats et plusieurs perspectives de recherche.

2. Démarche expérimentale

Notre démarche expérimentale est représentée Figure 1. La démonstration de l'universalité du jeu de la vie s'appuyant sur la présence de planeurs, un premier algorithme évolutionnaire est conçu pour trouver des automates de l'espace E acceptant des planeurs. Cet algorithme utilise une fonction d'évaluation spécifique prenant en compte l'évolution spatiale, le nombre de cellules vivantes ainsi que la présence de planeurs. Contrairement à celui du jeu de la vie la plupart des planeurs découverts ne peuvent se déplacer que dans un seul sens. Ces planeurs ne sont donc directement pas utilisables dans un schéma de démonstration d'universalité tel que celui exposé par Conway dans (Berlekamp *et al.*, 1982).

Nous avons donc recherché des automates acceptant des planeurs multidirectionnels. Pour cela, nous proposons un deuxième algorithme évolutionnaire assurant la découverte d'automates isotropes (*i.e.* pour lesquels si deux cellules ont les mêmes cellules voisines à l'état 1 par symétries alors ces deux cellules ont le même état à la génération suivante de l'automate). Cet algorithme a permis la découverte de nombreux automates acceptant des planeurs et nous avons remarqué la présence d'automates acceptant des canons à planeurs.

Nous avons montré l'universalité d'un automate parmi ceux découverts précédemment, que nous avons appelé R_0 , grâce à la présence d'un canon à planeurs. Afin de trouver d'autres automates universels, nous avons recherché des automates acceptant des canons à planeurs parmi ceux acceptant déjà des planeurs.

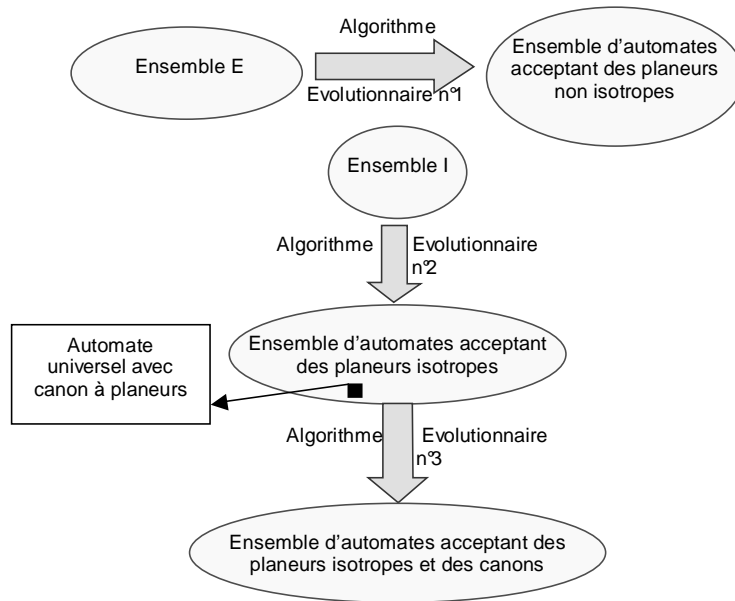


Figure 1 . Démarche expérimentale

3. Environnement de travail

Pour les automates de E , une cellule prendra l'état 0 ou 1 à la génération suivante en fonction de son état et de l'état de ses huit voisins. Sa fonction de transition est donc de $\{0,1\}^9$ dans $\{0,1\}$. L'ensemble $\{0,1\}^9$ étant de cardinalité 512, on peut coder les automates de E par des chaînes de 512 bits (Sapin, 2003). Chaque bit est associé à chacun des 512 voisinages possibles d'une cellule. Sa valeur est celle que prendrait une cellule ayant ce voisinage à la génération suivante de l'automate. La Figure 2 représente un automate de E . Le point à droite du voisinage pointé par la flèche indique que la cellule centrale de ce voisinage sera à l'état 1 à la génération suivante de l'automate.

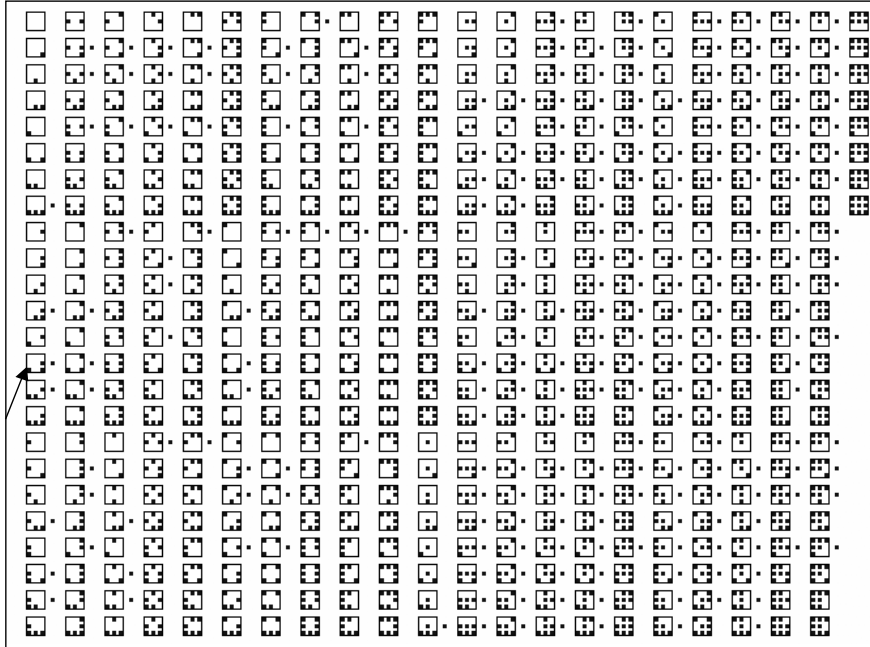


Figure 2 . Représentation d'un automate de E

Pour des raisons de concision et de lisibilité, une convention utilisant l'espace de Bays (Bays, 1987) a été adoptée pour la représentation des automates de E . Pour les automates de l'espace de Bays, une cellule passe de l'état 0 à 1 ssi elle a entre Fb et Fh cellules voisines à l'état 1 et une cellule reste à l'état 1 ssi elle a entre Eb et Eh cellules voisines à l'état 1. Les automates de cet espace, notées E_bE_h/F_bF_h , prennent donc en compte l'état d'une cellule et le nombre de ses cellules voisines à l'état 1 pour déterminer son état à la génération suivante. Cet espace, comportant 1296 automates, est donc un sous-espace de l'espace E . Un automate A de E sera noté :

$$(E_bE_h/F_bF_h) \{m_k\}_{0 < k \leq n}$$

où (E_bE_h/F_bF_h) est la restriction à l'espace de Bays de l'automate A , $\{m_k\}$ l'ensemble des modifications de A par rapport à cette restriction et n est le nombre de ces modifications. Les modifications correspondent à un bit qui change de valeur pour un voisinage. Ce voisinage est désigné par un numéro m_k correspondant à sa position dans la chaîne de 512 bits représentant l'automate A . Ce numéro est calculé en attribuant les poids représentés Figure 3 aux cellules vivantes.

128	64	32
16	256	8
4	2	1

Figure 3 . Poids des cellules utilisés dans le calcul du numéro d'un voisinage.

Par exemple, le numéro du voisinage fléché sur la Figure 2 est 13 ($1 + 8 + 4$).

Pour représenter un automate de l'ensemble des automates isotropes, appelé I , il suffit de représenter la valeur associée à chacun des 102 voisinages différents à des symétries près.

4. Recherche d'automates acceptant des planeurs

Ce paragraphe décrit l'algorithme évolutionnaire que nous avons utilisé pour rechercher de nouveaux automates de E capables d'engendrer des planeurs. Nous allons décrire l'algorithme utilisé, les résultats obtenus puis conclure par une synthèse de nos résultats.

4.1. Algorithme numéro 1

Nous allons décrire la fonction d'évaluation de notre algorithme, l'initialisation des individus puis les opérateurs évolutionnaires et enfin le déroulement de notre algorithme.

4.1.1. Fonction d'évaluation

Le calcul de la fonction d'évaluation est basé sur l'évolution, pendant P transitions, d'une configuration de cellules, générée aléatoirement dans un carré de côté C au centre d'un espace carré de côté U . Pendant cette évolution deux scores S_1 et S_2 sont calculés ainsi qu'un coefficient c . Après cette évolution, la présence de planeurs et de formes périodiques est testée.

4.1.1.1. Score S_1

A chaque transition, les dimensions du plus petit rectangle contenant toutes les cellules vivantes, appelé rectangle englobant, sont mesurées. En présence de planeurs, ce rectangle est de plus en plus grand au fil des générations. Nous avons donc voulu faire intervenir dans la fonction d'évaluation le fait que les dimensions de ce rectangle devaient grandir.

Pendant l'évolution de l'automate, l'algorithme a donc compté le nombre de fois n_1 où la surface de ce rectangle augmente et le nombre de fois n_2 où elle diminue. Pour un nombre L de générations de l'automate, on définit:

$$S_1 = (n_1 - n_2 + L) / L$$

Ce score S_1 représente la capacité de l'automate à s'étendre dans l'espace pour un nombre L de générations.

4.1.1.2. Score S_2

Pour calculer le score S_2 , le nombre total de cellules vivantes à chaque génération est pris en compte. Ce nombre, dans le jeu de la vie, tend à rester stable. On peut le constater Figure 4, où il est représenté conjointement avec la surface du rectangle englobant. Nous avons choisi de prendre en compte la stabilité du nombre de cellules vivantes dans notre fonction d'évaluation, cette stabilité étant vérifiée dans le jeu de la vie qui était le seul automate universel de E au moment où nous avons fait ce choix.

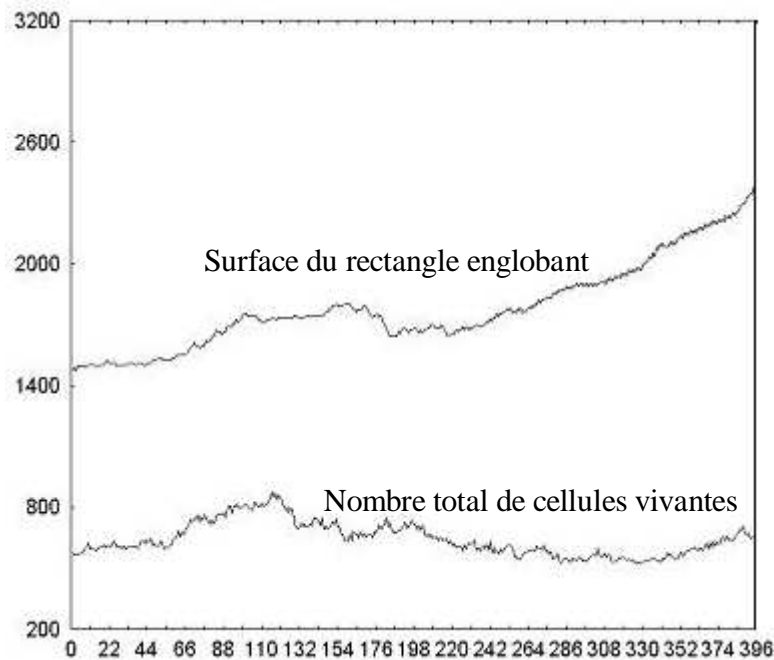


Figure 4 . Moyenne sur 100 tests indépendants pendant 400 générations de l'évolution au cours du temps de la surface du rectangle englobant et du nombre total de cellules pour le jeu de la vie.

La capacité à maintenir un nombre stable de cellules a été évaluée par le score S_2 pour un nombre L de générations. Ce score est défini sur la base du nombre m_2 de fois où le nombre total de cellules a augmenté et du nombre m_1 de fois où le nombre de cellules a diminué pendant un nombre L de générations de l'automate :

$$S_2 = (m_1 - m_2 + L) / L$$

Avec une première fonction d'évaluation $F_1 = S_1 * S_2$ nous avons observé une convergence vers des automates ayant des bons scores S_1 et S_2 mais n'acceptant pas de planeurs. Dans ces automates, l'augmentation du rectangle englobant était due à un déplacement global des cellules de l'automate dans une direction privilégiée. Pour pénaliser ce type de comportement, nous avons fait apparaître dans la fonction d'évaluation un coefficient reflétant le déplacement du centre de gravité des cellules vivantes.

4.1.1.3. Coefficient c

Le coefficient c traduit l'importance du déplacement du centre de gravité des cellules vivantes. Il est défini par $1 + (dg/sz)$, où dg est la distance euclidienne entre le centre de gravité et le milieu de l'univers carré de taille sz . La propension d'un automate à avoir un comportement attendu est alors évaluée par :

$$F_2 = S_1 * S_2 / c = ((n_1 - n_2 + L) * (m_1 - m_2 + L) / (cL^2))$$

Cette fonction d'évaluation permet l'émergence d'automates ne présentant pas de déplacement global des cellules dans une direction privilégiée. Malheureusement, les automates découverts ne permettent pas l'apparition spontanée de planeurs. Nous avons donc modifié la fonction d'évaluation de manière à favoriser ce comportement.

4.1.1.4. Détection de planeurs

La présence de planeurs a été détectée par le test suivant, inspiré du test de Bays (Bays, 1987) : chaque groupe de cellules connexes (cf. Figure 5) est isolé et placé dans un espace vide, puis son évolution est calculée sur S transitions.

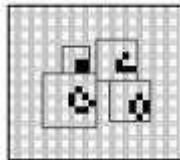


Figure 5 . Groupes de cellules isolées.

Pour chacune d'elle, la forme d'origine est recherchée dans l'univers de test.

Trois cas peuvent se produire :

- La forme initiale est réapparue à l'endroit où elle était à la première génération (elle est alors périodique),
- elle est réapparue à un autre endroit (la forme est alors un planeur),
- elle n'est pas réapparue (elle est alors en cours d'évolution).

Nous avons fait intervenir dans la fonction d'évaluation le nombre de planeurs pour favoriser les automates dans lesquels ils apparaissent. Nous avons fait aussi intervenir le nombre de formes périodiques mais avec un poids moins important que le nombre de planeurs. Ces automates ont été pris en compte car nous avons supposé que, s'ils acceptaient des formes périodiques, ils pouvaient évoluer vers des automates acceptant des planeurs, grâce à l'algorithme évolutionnaire. Le nombre d'apparitions d'un planeur v et le nombre d'apparitions d'une forme périodique p ont été intégrés de la manière suivante dans la fonction d'évaluation :

$$F_3 = (S_1 * S_2 / c) * (1 + v + p / 100) = ((n_1 - n_2 + L) * (m_1 - m_2 + L) / (cL^2)) * (1 + v + p / 100)$$

4.1.2. Initialisation

L'algorithme évolutionnaire fait évoluer une population de A automates. Pour diminuer le nombre de générations requises par notre algorithme évolutionnaire, nous avons choisi, après plusieurs essais, d'engendrer la population initiale de la manière suivante. Un automate a de l'espace de Bays défini dans (Bays, 1987) et dans le paragraphe 3 est choisi aléatoirement. La population initiale est constituée de $A/5$ occurrences identiques de a , $A/5$ variantes de a obtenues avec 1 mutation sur a , $A/5$ automates obtenus avec 2 mutations sur a , $A/5$ automates obtenus avec 3 mutations sur a et $A/5$ automates obtenus avec 4 mutations sur a .

4.1.3. Opérateurs évolutionnaires

Nous avons implanté un croisement simple à un point central. La première moitié d'une règle de transition d'un automate et la deuxième moitié d'une règle d'un autre automate sont concaténées pour former une troisième règle de transition d'un troisième automate. Nous avons fait ce choix car les 256 premiers bits de la règles de transition d'un automate sont ceux qui déterminent la naissance des cellules et les 256 suivants déterminent la survie ou la mort des cellules.

La mutation consiste à modifier un bit parmi les 512 bits de l'automate.

4.1.4. Evolution

Nous avons choisi de conserver les meilleurs individus et de leur appliquer le croisement et la mutation. Après l'initialisation des A automates, le cycle suivant est donc itéré :

- Evaluation des automates par la fonction d'évaluation.

- Conservation des X automates pour lesquels la valeur de la fonction d'évaluation est la plus grande.
- Création de Y automates en répétant Y fois le processus suivant :
 - Choix d'un automate avec équiprobabilité parmi les X automates conservés.
 - Application d'une mutation à cet automate.
- Création de $A-X-Y$ automates en répétant $A-X-Y$ fois le processus suivant :
 - Choix de deux automates avec équiprobabilité parmi les X automates conservés.
 - Croisement de ces deux automates.
- Création d'une nouvelle population de A automates avec les automates conservés, ceux obtenus par mutation et ceux obtenus par croisement.

4.2. Résultats

Parmi toutes les valeurs testées pour les paramètres, le plus grand nombre d'automates acceptant des planeurs fut trouvé pour les valeurs suivantes : nombre d'automates : $A = 50$, nombre d'automates conservés : $X = 20$, nombre d'automates mutés : $Y = 20$, côté du carré de cellules : $C = 40$, côté de l'univers : $U = 200$ et nombre de transitions : $P = 300$. Ces automates sont présentés dans ce paragraphe.

4.2.1. Nouveaux automates

4.2.1.1. Automate R_1

L'automate (12/33) {41 78 84 86 93 115 154 178 181 229 333 355 409 465}, noté R_1 , qui présente 14 mutations, a été obtenu en 59 générations. Il permet l'émergence de nombreuses formes stables et de formes périodiques dont quelques exemples sont représentés Figure 6.

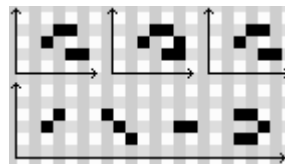


Figure 6 . Une forme périodique (en haut) représentée aux générations 0, 1 et 2 de gauche à droite et quatre formes stables acceptées par R_1 .

Nous avons représenté, Figure 7, une forme appelée en anglais « puffer » et que nous pouvons traduire ici par l'expression : *générateur de fumée*. Cette forme se déplace dans l'espace en laissant après son passage des formes stables ou

périodiques. Ce générateur de fumée se déplace de 12 cases horizontalement et 4 cases verticalement, il suit donc une pente de 1/3. Parcourant 12 cases en 40 transitions, ce générateur de fumée a une vitesse de 3/10 cellule par génération.

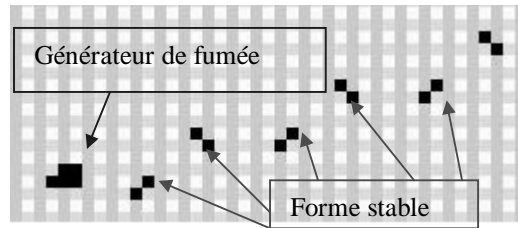


Figure 7 . Générateur de fumée accepté par R_1

4.2.1.2. Automate R_2

Un autre résultat a été obtenu en faisant évoluer l'automate 35/33 pendant 350 générations pour aboutir à l'automate (35/33) {15 29 39 50 57 69 132 138 163 194 294 306 316 326 331 338 360 366 374 402 432 441 450 461 474 483 485}, notée R_2 . Cet automate accepte plusieurs formes stables et périodiques (cf. Figure 8).

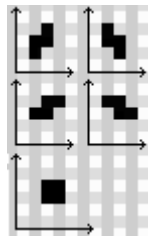


Figure 8 . Deux formes périodiques de période 2 représentés aux générations 0 et 1 de gauche à droite (en haut) et une forme stable (en bas) acceptées par R_2 .

Un planeur pouvant se déplacer horizontalement et de haut en bas suivant son orientation initiale est représenté Figure 9. Il a une période de 2 et une vitesse de $\frac{1}{2}$ cellule par génération. Notons que ce planeur existe dans l'automate 35/33 (Magnier *et al.*). Il apparaît toutefois à partir de toutes les configurations de cellules que nous avons testées pour l'automate R_2 , ce qui n'est pas le cas pour l'automate 35/33.

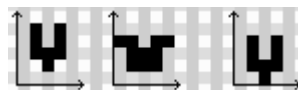


Figure 9 . Evolution d'un planeur accepté par R_2 sur une période.

4.2.1.3. Automate R_3

Un autre exemple est l'automate 22/33 {40 46 54 58 65 77 85 86 99 102 108 114 135 142 153 172 180 197 202 210 225 257 258 260 264 272 288 303 312 320 330 356 384 401 417 418 450 464}, notée R_3 , obtenu en 30 générations. Un planeur de période 11 est représenté Figure 10 pendant toute sa période et un deuxième planeur accepté par R_3 est représenté Figure 11.

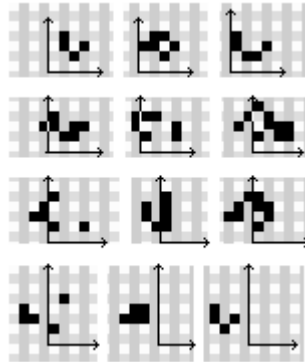


Figure 10 . Planeur accepté par R_3 de période 11 aux générations de 0 à 11 et de vitesse $5/11$ cellule par génération.

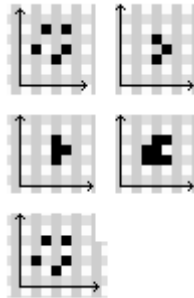


Figure 11 . Planeur accepté par R_3 de période 4 aux générations de 0 à 4 et de vitesse $1/4$ cellule par génération.

De nombreuses formes périodiques, telles que celles présentées Figure 12, existent aussi dans cet automate.

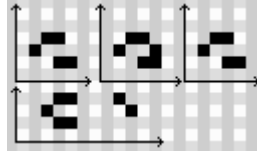


Figure 12 . Une forme périodique de période 2 (en haut) aux générations 0, 1 et 2 et deux formes stables acceptées par R_3 .

4.3. Synthèses

La plupart des planeurs découverts dans ce chapitre ne peuvent se déplacer que dans un seul sens. Ces planeurs ne sont donc pas directement utilisables dans un schéma de démonstration d'universalité tel que celui exposé par Conway dans (Berlekamp *et al.*, 1982). La mutation utilisée transforme un automate isotrope en un automate quelconque de E . Elle ne conserve donc pas l'isotropie des automates. Pour obtenir des planeurs isotropes, nous allons adapter l'opérateur de mutation dans le paragraphe suivant. L'algorithme évolutionnaire proposé assura donc l'isotropie des automates découverts.

5. Recherche d'automates isotropes acceptant des planeurs

Ce paragraphe décrit l'algorithme évolutionnaire que nous avons utilisé pour rechercher de nouveaux automates de I capables d'engendrer des planeurs. L'ensemble I étant un ensemble d'automates isotropes, (*i.e.* pour lesquels si deux cellules ont les mêmes cellules voisines à l'état 1 par symétries alors ces deux cellules auront le même état à la génération suivante de l'automate) les planeurs découverts seront donc isotropes. Nous allons décrire l'algorithme utilisé, les résultats obtenus puis terminer par une synthèse.

5.1. Algorithme numéro 2

Nous allons présenter la fonction d'évaluation de notre algorithme, l'initialisation des individus, les opérateurs évolutionnaires et enfin le déroulement de notre algorithme.

5.1.1. Fonction d'évaluation

Le calcul de la fonction d'évaluation est inspiré de celui de l'algorithme numéro 1. Il est basé sur l'évolution, pendant un temps P , d'une configuration finie de cellules de densité D . Cette configuration, appelée *soupe primordiale*, est

engendrée aléatoirement dans un carré de côté C (taille de la soupe) au centre d'un univers carré de côté U (taille de l'univers). Dans cet univers, évolue la soupe primordiale par la règle de transition de l'automate. Après cette évolution, le test suivant, inspiré du test de Bays (Bays, 87), est effectué : chaque groupe de cellules connexes est isolé afin de déterminer s'il s'agit d'un planeur, d'une forme périodique ou d'une autre forme. A cette fin, chaque cellule vivante fait l'objet d'un test, utilisant le paramètre R (rayon maximum d'un motif) représenté Figure 13.

```

N=2
Tant que (il existe au moins une cellule vivante
          espacée d'un rayon N de la cellule centrale)
          ET (N < R)
          N=N+1
Fin tant que

```

Figure 13. Test utilisé dans le calcul de la fonction d'évaluation de l'algorithme numéro 2.

La Figure 14 donne un exemple dans lequel certaines cellules sont espacées d'un rayon 2 de la cellule centrale (c.-à-d. traversées par le cadre fin) et aucune n'est espacée d'un rayon 3. La valeur de N étant de 3 à la fin de la boucle, on parlera de motif de rayon 3.

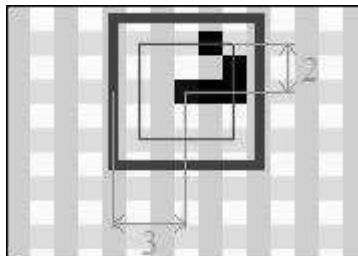


Figure 14. Illustration de la recherche d'un groupe de cellules connexes.

Chaque groupe de cellules connexes est ensuite placé dans un espace vide, de forme carrée et de côté T (taille de l'espace de test). Son évolution est alors calculée pendant un temps S (temps de test). A chaque transition, la forme d'origine est recherchée dans l'univers de test. Trois cas peuvent se produire :

- La forme initiale est réapparue à l'endroit où elle était à la première génération (elle est alors périodique),
- elle est réapparue à un autre endroit (la forme est alors un planeur),
- elle n'est pas réapparue (elle est alors en cours d'évolution).

La fonction d'évaluation est évaluée par le produit du nombre d'apparitions d'un planeur et du nombre d'apparitions d'une forme périodique.

5.1.2. Initialisation

La population initiale est constituée de A automates de I . Les bits de leurs fonctions de transition sont déterminés aléatoirement avec une probabilité de 0.5 pour les valeurs 0 ou 1.

5.1.3. Opérateurs évolutionnaires

La mutation est réalisée par choix aléatoire d'un voisinage avec équiprobabilité et modification des bits de ce voisinage et des bits des voisinages qui lui sont symétriques (Sapin, 2003). Sur la Figure 15, quatre voisinages symétriques sont représentés. Si le bit d'un de ces voisinage est modifié par la mutation les bits des trois autres voisinages le seront aussi. L'isotropie de l'automate est donc conservée.

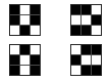


Figure 15. Quatre voisinages symétriques

Nous avons réalisé un opérateur de croisement simple à un point central fonctionnant de la même façon que celui de l'algorithme numéro 1.

5.1.4. Evolution

Le cycle d'évolution de l'algorithme numéro 1 est implémenté.

5.2. Résultats

Les valeurs des paramètres de notre algorithme évolutionnaire sont les suivantes : nombre d'automates : $A = 50$, côté du carré de cellules : $C = 40$, côté de l'univers : $U = 340$, rayon maximum : $R = 8$, côté de l'espace de test : $T = 20$, nombre de transitions : $P = 150$, densité de la configuration de cellules $D = 0,4$ et temps de test : $S = 6$. Ces valeurs ont été justifiées dans (Sapin *et al.*, 2004 a). Cette justification est basée sur des observations expérimentales et des mesures statistiques des apparitions de planeurs et de formes périodiques. Nous allons détailler les résultats sur les planeurs qui sont acceptés par les automates découverts puis décrire un automate découvert démontré universel.

5.2.1. Planeurs découverts

Chaque automate découvert accepte un ou plusieurs planeurs. Certains planeurs sont apparus plusieurs fois dans des automates différents. Nous avons représenté plusieurs planeurs ayant des propriétés particulières tels que les planeurs étant

apparus le plus souvent, ceux comportant le plus de cellules, ceux ayant la période la plus grande et des planeurs ayant des propriétés plus spécifiques.

5.2.1.1. Planeurs apparus le plus souvent

Nous avons représenté, Figure 16, les planeurs qui sont apparus le plus souvent dans les automates découverts, avec leur taux d'apparition. Ces planeurs sont constitués de trois cellules vivantes ce qui est le minimum pour des planeurs isotropes (deux cellules définissent une direction mais pas un sens).

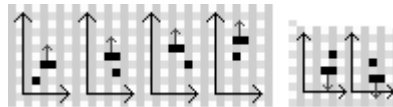


Figure 16 . Un planeur de période 4 et un de période 2 apparus dans respectivement 5.7% et 4.4% des automates découverts par notre algorithme évolutionnaire.

5.2.1.2. Planeurs comportant le plus de cellules

Les planeurs ont un nombre de cellules variable aux différentes générations de sa période. Nous avons choisi de nous intéresser aux nombres de cellules des planeurs découverts à des générations arbitraires qui sont celles pendant lesquelles ils sont découverts. La distribution des nombres de cellules des planeurs pendant ces générations est représentée Figure 17 pour des nombres de cellules allant de 1 à 11.

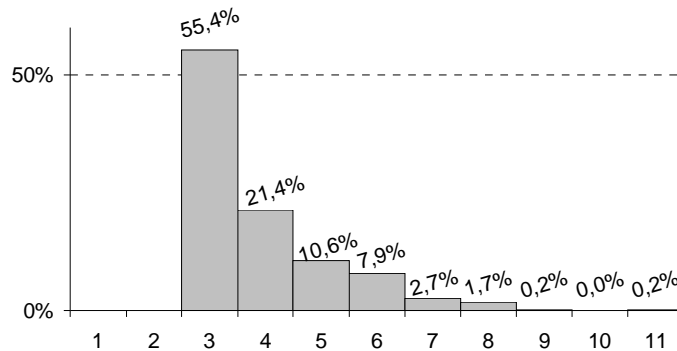


Figure 17 . Distribution des nombres de cellules des planeurs aux générations pendant lesquelles ces planeurs sont découverts.

On peut remarquer que plus de la moitié des planeurs découverts ont trois cellules. Ce résultat est en accord avec l'idée qu'une forme apparaît spontanément d'autant moins souvent qu'elle comporte beaucoup de cellules.

Parmi les planeurs apparus dans les automates découverts, celui constitué du plus grand nombre de cellules est représenté Figure 18.

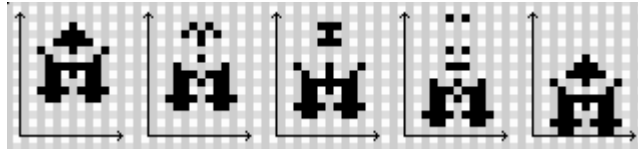


Figure 18 . Un planeur découvert par notre algorithme évolutionnaire représenté génération après génération.

Pendant les quatre générations de sa période, ce planeur, apparu qu'une seule fois, comporte successivement 42, 36, 41 et 37 cellules. De plus gros planeurs peuvent apparaître tel que celui représenté Figure 19 qui comporte jusqu'à 141 cellules, mais la valeur du rayon maximum d'un motif décrit paragraphe 5.1.1, qui est de 8, limite la taille des planeurs détectables par notre fonction d'évaluation.

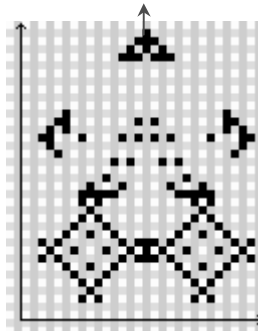


Figure 19 . Première génération d'un planeur apparaissant spontanément dans certains automates qui l'acceptent.

5.2.1.3. Période des planeurs

La distribution des périodes des planeurs découverts est représentée sur la Figure 20 pour des périodes allant jusqu'à 9.

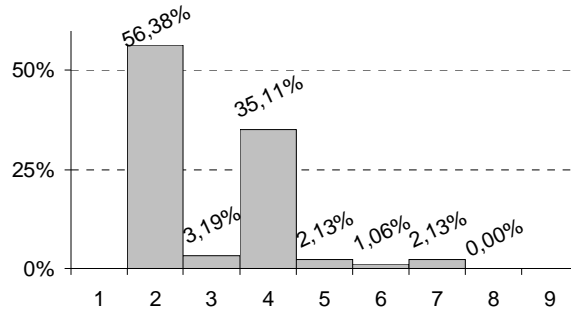


Figure 20 . Distribution des périodes des planeurs découverts.

La majorité des planeurs découverts de période 4 sont, à l’instar du planeur de gauche de la Figure 16, des planeurs dans lesquels les formes de la première et la troisième générations sont identiques par symétrie suivant l’axe de direction du planeur.

Parmi les planeurs découverts par notre algorithme, celui ayant la plus grande période est représenté Figure 21. Ce planeur se déplace d’une seule cellule durant sa période ce qui est le minimum. Ayant la plus grande période parmi les planeurs découverts et se déplaçant du minimum de cellules pendant celle-ci, Il a la vitesse la plus lente des planeurs découverts.

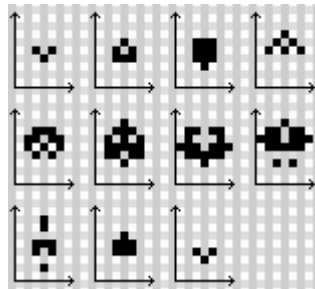


Figure 21 . Planeur découvert se déplaçant à une vitesse de 1/10 cellule par génération.

5.2.1.4. Planeurs ayant un déplacement particulier

Un planeur se déplace de x cellules horizontalement et y cellules verticalement. Pour la majorité des planeurs découverts, les valeurs x et y suivent une des deux propriétés suivantes :

- $x = 0$ ou $y = 0$, ce qui correspond à un déplacement suivant un axe.
- $x = y$, ce qui correspond à un déplacement suivant une diagonale.

Pour quelques planeurs découverts les valeurs x et y ne suivent aucune de ces deux propriétés. Nous en avons représenté un Figure 22 pour lequel $x=-3$ et $y=-1$.

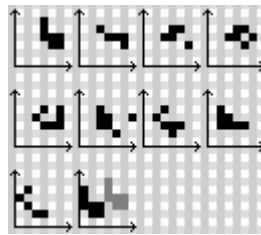


Figure 22 . Un planeur découvert par notre algorithme évolutionnaire avec la position du planeur à la première génération indiquée à la dernière génération.

Ce planeur est isotrope et nous avons représenté Figure 23 toutes les directions possibles qu'il peut prendre. Ce type de planeur, apparaissant rarement, a huit directions possibles. Aucun planeur de ce type n'est, à notre connaissance, accepté par le jeu de la vie.

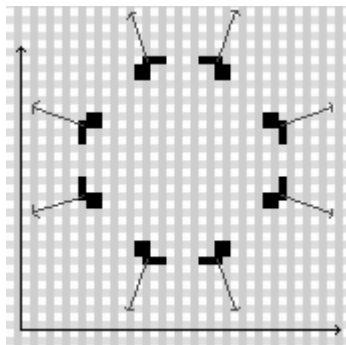


Figure 23 . Un planeur représenté dans huit positions différentes.

5.2.1.5. Planeurs apparus périodiquement

Dans certains automates, les planeurs qui émergent spontanément sont expulsés périodiquement par une forme, appelée canon à planeurs, apparue pendant l'évolution de l'automate. Une de ces formes est représentée Figure 24.

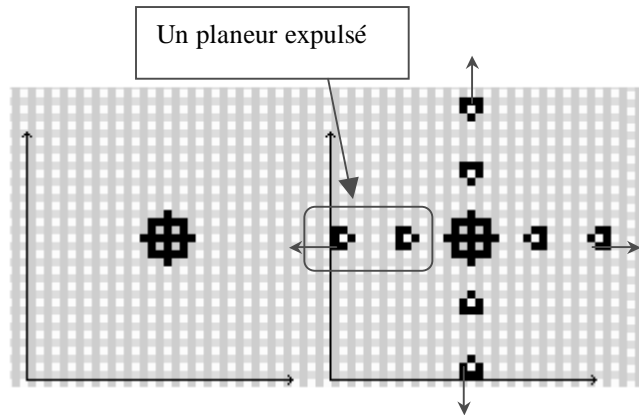


Figure 24 . Une forme expulsant des planeurs représentée aux générations 1 et 35 .

Dans un automate, l'apparition spontanée d'un canon à planeurs assure l'apparition de planeurs. Nous avons toutefois trouvé assez peu d'automates ayant cette propriété. Cette faible quantité d'automates trouvés permet de penser que ces types de forme émergeant spontanément sont plus rares que les planeurs.

5.2.2. Automate universel

Nous nous sommes intéressé à un automate choisi arbitrairement parmi ceux acceptant un canon à planeurs pour démontrer son universalité. Nous avons appelé R_0 cet automate représenté Figure 25.

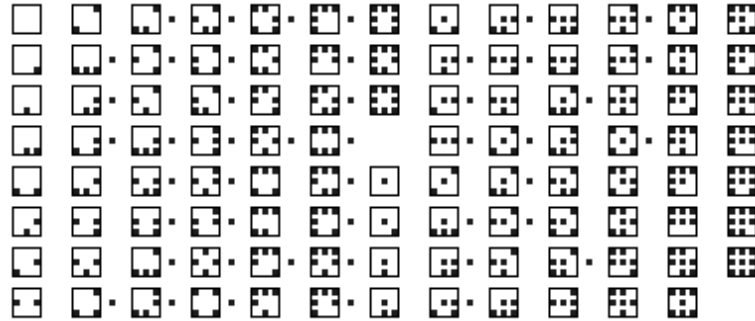


Figure 25. Représentation de l'automate R_0 .

La simulation d'une porte *ET* par le jeu de la vie décrite dans (Berlekamp *et al.*, 1982) nécessite un canon à planeur mais aussi une forme, appelée en anglais « eater » et que nous pouvons traduire par *mangeur*. Nous avons donc recherché un automate acceptant un mangeur parmi les automates qui acceptent le canon à planeurs de R_0 (Sapin *et al.*, 2003). Nous avons trouvé un automate R qui diffère peu de R_0 . Nous avons représenté le canon à planeurs ainsi que le mangeur Figure 26.

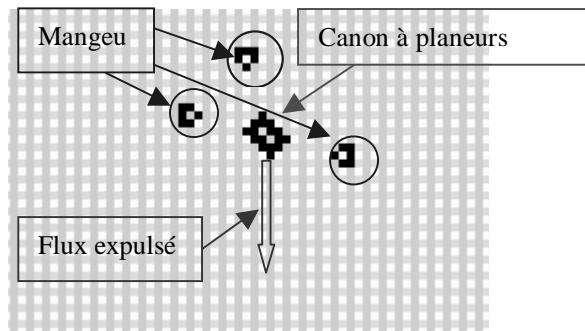


Figure 26 . Le canon à planeurs de R à la première génération.

Grâce au canon et au mangeur nous avons simulé une cellule du jeu de la vie avec l'automate R . Cette simulation est représentée Figure 27.

Pour montrer l'universalité de l'automate R nous avons montré qu'il était possible de réaliser une simulation du jeu de la vie par R . Des carrés contiguës et identiques répétés périodiquement et contenant chacun une simulation d'une cellule réalise un pavage du plan du jeu de la vie. La difficulté est de connecter chaque carré à ces de ces huit voisins numérotés de C_1 à C_8 .



Figure 27 . Simulation d'une cellule du jeu de la vie par l'automate R.

Ces interconnexions sont représentées Figure 28. Sur cette figure la cellule simulée S est représentée ainsi que les flux de planeurs de C_1 à C_8 représentant les

valeurs des huit cellules voisines. Trois styles de trait différents distinguent la fonction des différents flux. La valeur S de la cellule simulée doit être « envoyée » vers ses huit voisins et vers le calcul de sa propre valeur pour la génération suivante. Les parcours des flux réalisant cette tâche sont représentés en trait continu. Les flux permettant à la cellule simulée de « récupérer » les flux « envoyés » par les cellules voisines sont représentés en trait pointillé. Les traits du dernier style sont ceux qui permettent à des cellules diagonalement adjacentes d'être connectées. A chaque flux en entrée à gauche correspond un flux en sortie à droite, et de même pour les autres côtés, ce qui permet un pavage du plan avec des simulations identiques.

Le jeu de la vie étant universel l'automate R est donc lui aussi universel (Sapin, 2003).

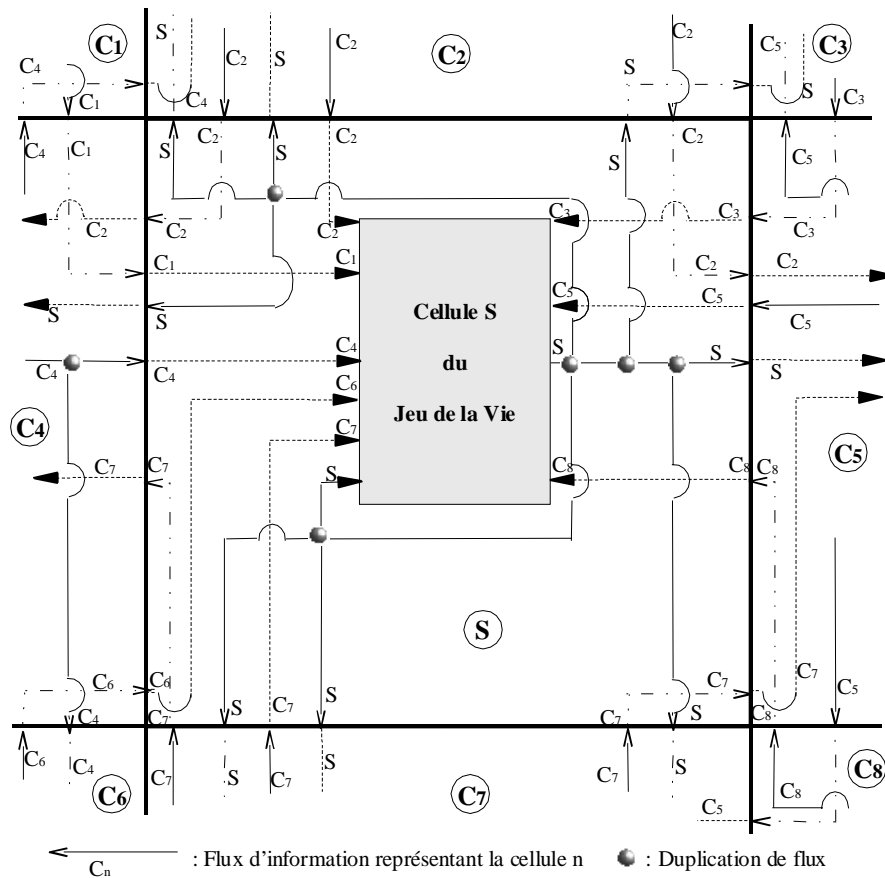


Figure 28 . Interconnexion des simulations des cellules du jeu de la vie.

5.3. *Synthèse*

Nous avons conçu un algorithme évolutionnaire capables de faire émerger de nombreux automates cellulaires isotropes acceptant des planeurs. Pour atteindre cet objectif nous avons proposé un opérateur de mutation qui conserve l'isotropie des automates.

La recherche de planeurs nous a permis de découvrir quelques automates acceptant des canons à planeurs. Nous avons démontré l'universalité d'un de ces automates. Pour découvrir d'autres automates universels nous avons donc choisi de rechercher des canons à planeurs acceptés par les automates découverts.

6. Recherche de canons à planeurs

Ce paragraphe décrit l'algorithme évolutionnaire que nous avons utilisé pour rechercher des canons à planeurs acceptés par les automates découverts par l'algorithme numéro 2. Nous allons décrire l'algorithme utilisé, les résultats obtenus puis terminer par une synthèse des résultats.

6.1. *Algorithme numéro 3*

Nous allons présenter la fonction d'évaluation de notre algorithme et l'initialisation des individus. Nous allons décrire ensuite les opérateurs évolutionnaires et le déroulement de notre algorithme.

6.1.1. *Fonction d'évaluation*

Le calcul de la fonction d'évaluation suit la même procédure que celui de l'algorithme numéro 2. Une configuration aléatoire de cellules évolue pendant plusieurs générations. Après cette évolution, chaque groupe de cellules connexes est isolé et évolue seul dans un espace vide. A chaque génération, la forme d'origine est recherchée. Trois cas peuvent se produire :

- La forme initiale est réapparue à l'endroit où elle était à la première génération, dans ce cas la forme initiale est effacée et les formes restantes sont examinées :

- Les formes restantes sont des planeurs : un examen visuel déterminera alors si la forme initiale est un canon.

- Les formes restantes ne sont pas que des planeurs : la forme initiale est alors en cours d'évolution.

- La forme initiale est réapparue à un autre endroit (cette forme est alors un planeur).

- Elle n'est pas réapparue (elle est alors en cours d'évolution).

La fonction d'évaluation est, à présent, évaluée par le nombre d'apparitions d'un planeur divisé par le nombre total de cellules vivantes après le temps d'évolution.

6.1.2. Initialisation

La population initiale est à présent constituée de A occurrences d'un automate obtenu avec l'algorithme numéro 2.

6.1.3. Opérateurs génétiques

La mutation est réalisée par le choix d'un voisinage et la modification des bits de ce voisinage et des bits des voisinages qui lui sont symétriques. Afin de guider la recherche de l'algorithme, les automates mutés pour lesquels le planeur de l'automate d'origine n'est plus accepté ne sont pas conservés et un autre voisinage est choisi.

Nous avons réalisé un opérateur de croisement simple à un point central fonctionnant de la même façon que celui de l'algorithme numéro 2.

6.1.4. Evolution

Le cycle d'évolution de l'algorithme numéro 1 est implémenté.

6.2. Résultats

Nous avons testé les tailles A de population 10, 20, 50, et 100 en faisant évoluer 100 fois notre algorithme évolutionnaire pendant respectivement 500, 250, 100, et 50 générations. Sur 100 évolutions, le nombre de fois où un canon a été trouvé pour chaque taille de population est indiqué Figure 29. Le nombre de fois où un canon est trouvé diminue pour une taille de population allant 50 à 100 car le nombre de générations n'est plus assez important pour qu'un canon apparaisse.

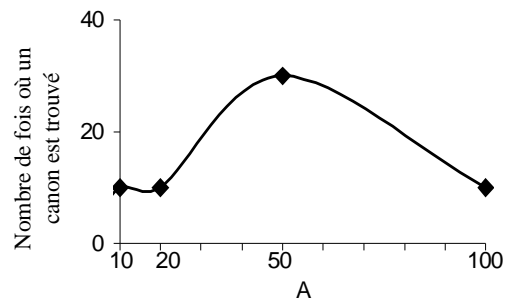


Figure 29 . Nombre de fois où un canon à planeurs est découvert par notre algorithme évolutionnaire sur 100 évolutions pour des tailles A de population de 10, 20, 50, et 100.

Nous avons représenté un canon découvert de période 19 sur la Figure 30. Ce canon expulse un premier planeurs au premier quart de sa période à la génération

6. Au milieu de sa période, génération 10, ce canon est dans la même forme qu'au début de sa période après une rotation d'angle π . Il expulse donc un deuxième planeur à la génération 15.

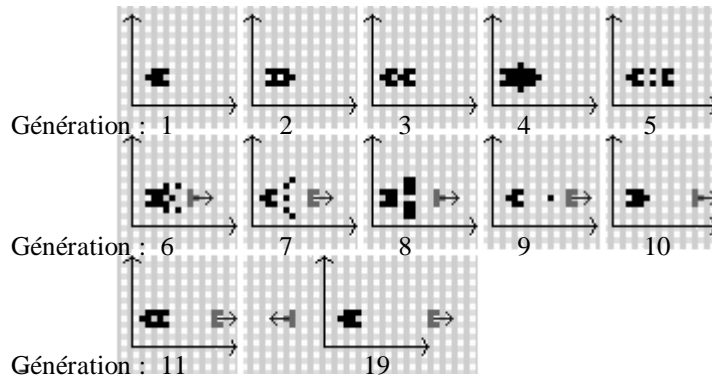


Figure 30 . Canon à planeurs, découvert par algorithme évolutionnaire, avec les planeurs en clair et leur direction indiquée par les flèches.

La Figure 31 présente un deuxième canon à planeurs découvert de période 39 qui expulse à la génération 37 deux planeurs.

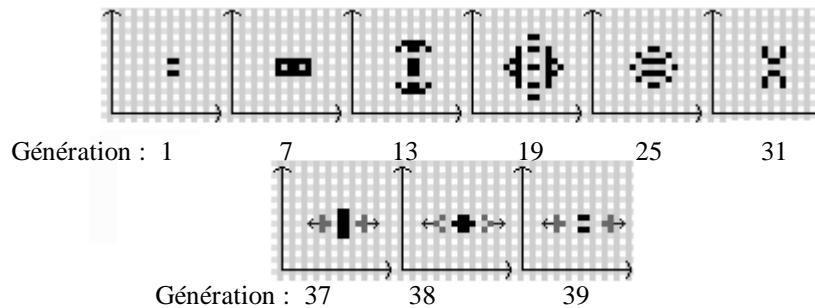


Figure 31 . Canon à planeurs, découvert par algorithme évolutionnaire, avec les planeurs en clair et leur direction indiquée par les flèches.

6.3. Synthèse

Nous avons conçu un algorithme évolutionnaire capables de découvrir, à partir des automates cellulaires découverts par l'algorithme numéro 2, des automates acceptant des canons à planeurs.

Pour atteindre cet objectif, nous avons proposé un opérateur de mutation qui conserve un planeur dans ces automates. Pendant le calcul de la fonction d'évaluation nous avons implémenté une détection automatique de canons à planeurs.

Nous avons découvert de nombreux automates cellulaires acceptant des canons à planeurs.

7. Conclusion

Nous nous sommes intéressé au problème, liée à l'émergence de la calculabilité dans les systèmes complexes, de la fréquence des automates cellulaires universels. Nous avons donc donné des éléments de réponse au problème posé par Wolfram dans (Wolfram, 1983) : « How common are computational universality and undecidability in cellular automata ».

Notre stratégie a consisté à rechercher, par algorithmes évolutionnaires, des automates cellulaires acceptant des formes particulières. Nous avons d'abord conçu un premier algorithme évolutionnaire qui nous a permis de découvrir de nombreux automates 2D, à deux états, acceptant des planeurs. La plupart des planeurs découverts ne pouvaient se déplacer que dans un seul sens. Ces planeurs n'était donc pas directement utilisables dans un schéma de démonstration d'universalité tel que celui exposé par Conway dans (Berlekamp *et al.*, 1982). Nous avons donc adapté notre algorithme pour qu'il fasse émerger des automates cellulaires acceptant des planeurs isotropes. La recherche de planeurs nous a permis de découvrir quelques automates acceptant des canons à planeurs. Nous avons démontré l'universalité de l'un de ces automates, appelé *R*. Pour découvrir d'autres automates universels nous avons donc choisi de rechercher des canons à planeurs acceptés par les automates découverts. Nous avons découvert de nouveaux automates cellulaires acceptant des canons à planeurs. La méthode proposée ici a donné des résultats prometteurs et peut être généralisé à d'autres automates auxquels on peut appliquer le schéma de démonstration de l'universalité de *R*.

Cette généralisation peut permettre la découverte de nombreux automates universels connus ou encore inconnus. Cette perspective peut être réalisée par une automatisation complète du processus de recherche d'automates cellulaires universels que nous avons proposé. Cette automatisation permettra de concevoir un algorithme évolutionnaire découvrant de nouveaux automates cellulaires universels.

La découverte d'un grand nombre d'automates universels peut permettre d'élaborer une nouvelle classification des automates cellulaires, plusieurs classifications existantes étant déjà basées sur les notions d'universalité et de

complexité (Wolfram, 1984)(Gutowick *et al.*,1988). D'autre part la connaissance d'un nombre important d'automates universels pourrait permettre de comparer les règles de ces automates pour chercher des similitudes entre elles et caractériser les paramètres, s'ils y existent, permettant l'émergence de comportements complexes globaux tel que l'universalité.

Bibliographie

- Bays C., « Candidates for the game of life in three dimensions », *Complex Systems*, vol. 1, 1987, p. 373-400.
- Banks, E.R., « Universality in Cellular Automata », *IEEE 11th Annual Symposium on Switching and Automata Theory*, 1970, p. 194-215.
- Berlekamp E. R., Conway J. H., Guy R. K., *Winning Ways for your mathematical plays* vol. 2, New York, Academic press, 1982.
- Codd E. F., *Cellular Automata*, Academic Press, 1968.
- Drossel B., Schwabl F., « Self organized criticality in a forest-fire model », *Physica A*, vol. 191, 1992, p. 47-50.
- Epstein I. R., « spiral Waves in Chemistry and Biology », *Science*, vol. 252, 1991, p. 67.
- Ermentrout, G. Lotti, and I. Margara , « Cellular Automata Approaches to Biological Modeling », *Journal of Theoretical Biology*, vol. 60, 1993, p. 97-133.
- Gardner M., « The fantastic combinaisons of John Conway's new solitaire game “ Life “ », *Scientific American*, 1970.
- Gardner M., « On Cellular Automata, Self-reproduction, the Garden of Eden, and the Game of Life », *Scientific American*, vol. 224, 1971, p. 112-118.
- Gutowitz H., Langton. C., « Methods for designing 'interesting' cellular automata », *CNLS News Letter*, 1988.
- Lindgren K., Nordahl M., « Universal computation in simple one dimensional cellular automata », *Complex Systems*, vol. 4, 1990, p.299-318.
- Magnier M, Lattaud C, Heudin J.C., « Complexity classes in the Two-dimensional Life Cellular automata Subspace », *Complex Systems*, vol. 11, 1997.
- Margolus N. *Physics and Computation*. Thèse de Doctorat, MIT, 1988.
- Morita K., Tojima Y., Katsunobo I., Ogiro T., *Universal Computing in Reversible and Number-Conserving Two-Dimensional Cellular Spaces* , Collision-Based Computing, Andrew Adamatzky (ed.), Springer Verlag, 2002.
- Von Neumann J., Burks A., *Theory of Self-Reproduction Automata*, University of Illinois Press, 1966, p. 77.
- Oda S. H., Kiura K., and Ueda K., *The Application of Cellular Automata to Network Externalities in Consumer's Theory : A Generalisation of Life Game*, Langton, C. G. and Shimohara K. (ed), MIT Press, 1977, p. 473-480.

- Rendell P., *Turing Universality of the Game of Life*, Collision-Based Computing, Andrew Adamatzky (ed.), Springer Verlag, 2002.
- Sapin E., Recherche par algorithmes évolutionnaires d'automates cellulaires universels, Thèse de doctorat, Université de Bourgogne, 2003.
- Sapin E., Bailleux O., Chabrier J.J. « Research of a cellular automaton simulating logic gates by evolutionary algorithms », *EuroGP2003, Lecture Notes in Computer Science*, vol. 2610, 2003, p. 414-423.
- Sapin E., Bailleux O., Chabrier J.J. « Research of complex forms in the cellular automata by evolutionary algorithms », *EA03, Lecture Notes in Computer Science*, vol. 2936, 2004, p. 373-400. a
- Sapin E., Bailleux O., Chabrier J.J., Collet P., « A New Universal Cellular Automaton Discovered by Evolutionary Algorithms », *GECCO2004, Lecture Notes in Computer Science*, vol. 3102, 2004, p. 175-187. b
- Turing A., *Intelligent Machinery*, report for National Physical Laboratory, 1948.
- Wolfram S., « Statistical Mechanics of Cellular Automata », *Reviews of Modern Physics*, vol. 55, 1983, p. 601-644.
- Wolfram S., « Universality and complexity in cellular automata », *Physica D*, vol. 10, 1984, p. 1-35.
- Wolfram S., « Twenty Problems in the Theory of Cellular Automata », *Physica Scripta*, vol. 9, 1985, p. 170-183.
- Wolfram S., « Random Sequence Generation by Cellular Automata. Advances in Applied Mathematics », vol. 7, 1986, p. 123-169.
- Wolfram S., Packard N. H., « Two-Dimensional Cellular Automata », *Journal of Statistical Physics*, vol. 38, 1985, p. 901-946.

Emmanuel Sapin a passé sa thèse à l'université de Bourgogne et enseigne actuellement dans l'établissement scolaire St Joseph à Dijon. Ses travaux, effectués pendant sa thèse et depuis celle-ci, se situent aux niveaux de l'étude des systèmes complexes et de l'approche évolutionniste de la résolution de problèmes. Ses centres d'intérêt incluent notamment l'informatique fondamentale et théorique, la logique, la complexité, le calcul formel, l'algorithmique combinatoire, la programmation par contraintes, les systèmes multi-agents ou la bio-informatique.