

## TITRE

Merwan Achibet

## Résumé

abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract ab-  
stract abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract  
abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract  
abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract  
abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract  
abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract

## 1 Introduction

## 2 État de l'art

La modélisation de systèmes complexes est longtemps uniquement passée par l'usage de méthodes mathématiques; typiquement, des systèmes d'équations différentielles. Ces techniques permettent de décrire des lois d'évolution et d'observer, ainsi que de prédire par extrapolation, le comportement de phénomènes du réel. Dans le cas de systèmes prenant en compte un vaste jeu de paramètres, cette approche peut néanmoins se révéler délicate à employer. Plus intrinsèquement, même si une telle modélisation est basée sur des observations ancrées dans la réalité, il s'agit d'une représentation conceptuelle d'un problème et aucune mimique des mécaniques sous-jacentes ne s'opère.

D'un point de vue historique, les prémices de l'informatique moderne et d'un tout autre paradigme de modélisation sont à attribuer aux esprits du milieu du vingtième siècle. Alan Turing introduit en 1936 la machine éponyme qui, bien que purement théorique, possède un module de contrôle ainsi qu'une mémoire et peut donc exécuter toutes sortes de procédures. Cette démarche se démarque de l'approche mathématique et semble plus humaine ; on ne résout pas un problème en utilisant des fonctions associant une quantité à un résultat mais on agit véritablement sur ses données. L'idée de base de Turing était d'ailleurs d'assimiler le fonctionnement de sa machine au travail d'une personne remplissant les cases d'un tableau infini CITA-

TION.

Entraînée par cette mouvance procédurale et en réaction aux réseaux de neurones de McCulloch et Pitts, John von Neumann introduit le terme *automate* en 1946. Simplement, un automate est une machine qui opère sur des données fournies en entrée en fonction de règles internes prédéfinies. On choisit de se concentrer sur une sous-catégorie d'automates, les automates finis à états, changeant leur représentation interne selon des règles de transition. John von Neumann et Stanislaw Ulam joignent leurs travaux pour concevoir l'automate cellulaire : un système comprenant un ensemble d'automates à états spatialement localisés (typiquement sous forme de grille) et interconnectés en fonction de leur proximité. Les entrées de chaque automate correspondent alors aux états des automates voisins et de cette organisation se dégagent de fortes relations d'interdépendance. Le jeu de la vie de Conway en est un exemple classique. La simplicité de ses règles, mise en contraste avec la variété des configurations engendrées, témoigne de la richesse des automates cellulaires CITA-TION.

Les automates cellulaires ont depuis été extensivement étudiés et sont appliqués à de nombreux phénomènes biologiques, physiques et sociaux [1]. La motivation d'Ulam lors de leur conception était d'ailleurs de modéliser la croissance de cristaux. On peut aussi citer en exemple la simulation de la dynamique de fluides [2], de la croissance de tumeurs [3], de l'évolution d'épidémies [4], de la ségrégation de population CITATION-SCHELLING. Leur caractère spa-

tial laisse supposer qu'ils sont particulièrement adaptés aux applications géographiques, et plus précisément, urbaines. Ils ne furent paradoxalement pas immédiatement exploités à cet effet et c'est seulement suite à un article de Waldo Tobler, en 1975, que le rapprochement entre les automates cellulaires et le domaine de la géographie apparaît clairement [5].

Une idée très exploitée dans ce domaine est d'associer un potentiel de transition à chaque cellule et ce, vers tous les états qu'elles peuvent prendre. Dans les modèles déterministes, la transition vers l'état à plus haut potentiel est appliquée tandis que dans les modèles stochastiques, un tirage aléatoire biaisée est préféré. Le potentiel d'une cellule à passer à un nouvel état est déterminé en fonction de paramètres propres à la simulation. Peuvent être pris en compte l'élévation du terrain, la densité de population, la proximité d'axes routiers, la proximité de centres urbains, l'âge des parcelles, leur valeur ; en fait, toute combinaison d'attributs relatifs à un réseau urbain. Par exemple, dans une simulation représentant les différents types d'usage, le passage d'une cellule à l'état *résidentiel* pourrait dépendre de la proximité des commerces et des routes et de l'éloignement des zones industrielles. Bien sûr, un nombre élevé de paramètres à prendre en compte requiert un couplage fin et l'impact de chaque variable peut être pondéré. Puisque les variations individuelles de paramètres n'émergent pas de manière transparente à la surface de la simulation, les modèles urbains basés sur des automates cellulaires doivent être finement calibrés et leur réalisme est un défi en soi. Pour contourner ce problème, Yeh et Li prône l'usage d'un réseau de neurones pour pondérer chaque paramètre à partir de l'analyse de données cartographiques historiques [6].

Il est important de noter que la simplicité des règles régissant le fonctionnement d'un automate cellulaire strict s'oppose à la qualité de la simulation, notamment dans le cadre de modèles spécifiques [7]. Dans ce cas, une prise de liberté quant aux formalisme originel est autorisée, voire nécessaire, pour obtenir des résultats satisfaisants [8].

La première limite que le formalisme cellulaire de base impose est la discrétisation des états que chaque cellule peut adopter. Même si

cette caractéristique fait partie intégrante des particularités qui confèrent aux automates cellulaires leur simplicité, la description de quantités pouvant arborer un éventail infini de valeurs est alors impossible. Plus concrètement, il est aisé de catégoriser les cellules d'un espace selon le fait, par exemple, qu'elles contiennent des installations humaines ou non (état booléen) [9, 10] ou de façon plus sophistiquée, en fonction de leur type d'usage (*résidentiel*, *commercial* et *industriel* [11] et plus [12]). Représenter des quantités réelles est moins aisé. Pour symboliser la densité au cœur d'un ensemble urbain, Semboloni utilise par exemple un automate cellulaire de dimension trois dans lequel plus une pile de cellules occupées est haute et plus la zone représentée est peuplée [13]. Plus généralement, on peut s'autoriser à représenter l'état d'une cellule par un vecteur contenant des valeurs réelles ; des règles de transitions adaptées et mesurées sont alors à mettre en place.

L'homogénéité d'un automate cellulaire fait partie intégrante de sa définition originelle : en mettant de côté l'état qu'elles adoptent, toutes les cellules sont identiques en forme et en structure de voisinage. Dans le cadre de notre problématique, cette approche est limitante car, dans une ville, les parcelles ne sont que rarement identiques et alignées. Similairement, la notion de voisinage est clairement à redéfinir. Pour des problèmes classiques, les voisinages de von Neumann et de Moore sont régulièrement utilisés mais la relation par contiguité qu'ils décrivent ne convient pas à la représentation des liens de dépendance à plus grande échelle se développant dans un système urbain. Le positionnement d'un bâtiment résidentiel dans une ville se base évidemment sur le voisinage direct des zones envisagées (on veut ajouter une maison dans un quartier résidentiel) mais il faut aussi prendre en compte les alentours plus distants (la centrale thermique se trouvant à 500 mètres du site peut poser problème). Une solution possible est d'étendre les voisinages de von Neumann et de Moore tout en conservant leur forme caractéristique. + AUTOMATE CELLULAIRE GRAPHE (THESE O'SULLIVAN) [14, 15].

Une prise de liberté quant à l'aspect temporel est aussi envisageable. Un automate cellulaire strict est synchrone, i.e. les changements d'état

de toutes les cellules s'effectuent simultanément. Si le choix était fait de mettre à jour chaque état de façon asynchrone, le comportement de l'automate en serait lourdement modifié. Par exemple, les qualités auto-réplicatives de certaines entités du jeu de la vie ne seraient pas garanties. Il est pourtant légitime de se questionner sur la validité d'un tel choix dans une simulation urbaine, premièrement parce qu'une ville est un système complexe et désordonné, deuxièmement parce les processus qui s'y déroulent sont réglés sur différentes échelles temporelles.

Les automates cellulaires ne sont pas l'unique moyen de modéliser la croissance urbaine. Plusieurs simulations existantes sont des systèmes multi-agent [16, 17]. Dans ces cas, un agent est assimilé à un promoteur immobilier et peut acheter des terres, les vendre, les développer ou changer leur type. Les actions qu'il entreprend sont évaluées en fonction de l'impact sur la ville (changement de la valeur immobilière, avis de la population) et des réglementations locales afin d'éviter toute configuration illégale.

D'autres solutions s'éloignant des systèmes complexes et penchant du côté de la génération procédurale de contenu existent. Souvent, le domaine d'application de telles méthodes est l'infographie, le cinéma et le jeu vidéo et l'objectif est alors de construire de manière automatique une ville visuellement réaliste sans se soucier de son caractère fonctionnel. Usuellement, la première étape est de générer un réseau viaire complet puis de placer le bâti en subdivisant récursivement les niches vides formées par les routes. Dans Citygen [18], un point  $p$  de l'espace est aléatoirement choisi puis on calcule un ensemble de plusieurs routes raccordant  $p$  au réseau routier existant en faisant varier leur déviation angulaire et un paramètre de bruit ; la route finale est celle pour laquelle la variation d'altitude est la plus faible. CityEngine [19] utilise un L-System dont les règles permettent de reproduire les différents motifs quadrillés, radiaux et organiques que l'on trouve dans une ville. La nature récursive des L-Systems permet à ces motifs de se combiner et d'apparaître à différents niveaux de profondeur ; les résultats sont saisissants. Dans une autre simulation [20], le tracé des routes suit les *hyperstreamlines* formées par un champ de vecteurs. Ce champ est calculé par com-

binaison de plusieurs autres champs de vecteurs, chacun représentant des contraintes directionnelles particulières telles que les zones interdites (eau, espaces verts), l'altitude et la densité de population. Ces techniques sont intrinsèquement géométriques, et comme précisé plus haut, le résultat est purement visuel, mais elles représentent une source d'inspiration à ne pas négliger.

Bien que les automates cellulaires soient couramment utilisés pour simuler le trafic routier (dans leur version 1D CITATION ou 2D [21]), ils s'accordent peu avec la construction même d'un réseau viaire. Dans les simulations cellulaires urbaines, le positionnement des routes a un impact sur le développement des cellules puisque le viaire attire le bâti mais le réseau est souvent fourni en entrée et reste fixe. Nous sommes amenés à nous interroger sur la capacité des automates cellulaires à modéliser le développement routier. Les relations de proximité les caractérisant sont-elles adaptées à la construction de structures dont l'échelle est celle de la ville et non plus celle de la parcelle ? Est-il bénéfique de représenter une route par un ensemble de cellules ou plutôt par une entité unique ?

En dehors de la sphère des automates cellulaires, quelques techniques procédurales de génération de routes ont été décrites plus haut. Dans les modèles à base d'agents, une solution est de mettre en place, en plus des agents promoteurs, deux types d'agents traceurs. Les *extenders* parcourent toute la surface du terrain à la recherche de bâtiments isolés puis tracent une route jusqu'au réseau urbain. Les *connectors* se déplacent uniquement sur le réseau viaire et y raccordent les bâtiments non connectés se trouvant dans leur rayon de détection [16]. PLUS ?

L'un des seuls modèles gérant à la fois l'évolution du réseau viaire et du bâti est présenté par Weber [22] et n'emploie pas d'automate cellulaire. Le principe est le suivant : à chaque agrandissement du réseau urbain, on crée plusieurs routes virtuelles en suivant des règles géométriques précises (allongement des voies existantes, limitation du degré des carrefours à 4, l'angle entre chaque rue tend vers 90 degrés). Parmi les  $n$  routes générées, une seule sera construite. Pour la choisir, le trafic sur ces nouvelles routes est simulé et l'on identifie celle

qui sera la plus bénéfique au réseau. AUTRES PAPIERS?

## 3 Structure et construction

### 3.1 Structure

Comme énoncé précédemment, les automates cellulaires sont des structures versatiles et puissantes dont le formalisme originel impose néanmoins quelques limitations; la principale étant, à nos yeux, un maillage figé et statique.

Pour répondre à notre problématique, il est nécessaire d'employer une structure souple et dynamique permettant de représenter à la fois le bâti et le viaire. On propose à cet effet d'utiliser le diagramme de Voronoï.

#### GROSSE DEFINITION

La vue d'un diagramme de Voronoï évoque naturellement le maillage de la ville, où chaque parcelle de terrain est représentée par une cellule et les parcelles voisines sont en contact. Ces frontières, prenant la forme des arêtes du diagramme de Voronoï, sont d'une importance capitale puisqu'elles correspondent aux espaces entre les bâtiments, et donc aux tracés des routes potentielles. Chaque cellule de Voronoï représente donc l'espace d'une parcelle tandis que chaque arête est une voie potentiellement constructible.

Les parcelles dont les cellules sont en contact (partageant une arête) sont considérées comme voisines et, à la manière des automates cellulaires, l'état d'une parcelle dépend de l'état des ses voisines.

Avant d'aller plus loin et de détailler la construction d'une telle structure et les règles régissant son évolution, il nous faut préciser certaines particularités.

Premièrement, l'espace qu'une cellule occupe ne représente absolument pas sa couverture cadastrale. On admet que l'empreinte du bâtiment est globalement contenue dans cette dernière sans aucune garantie qu'elle n'en déborde pas ou, à l'opposé, que la taille de la parcelle soit du même ordre de grandeur que l'aire de sa cellule.

#### IMAGE

Deuxième point important : les arêtes de Voronoï qui forment le réseau routier potentiel portent uniquement des informations topologiques et ne traduisent aucunement le tracé exact des routes qu'elles représentent. Autre-

ment dit, une telle arête décrit qu'une voie passe entre deux bâtiments précis sans pour autant renseigner sur sa direction ou sa courbure.

#### IMAGE

### 3.2 Construction

Notre système urbain est décrit sur deux niveaux; plus précisément, par deux graphes. On distingue donc le graphe parcellaire du graphe viaire car, bien qu'ils soient étroitement liés, ils représentent des couches différentes du réseau urbain.

#### 3.2.1 Le graphe parcellaire

Le graphe parcellaire correspond à la couche bâti. Ses nœuds sont les centres des parcelles et chacune des ses arêtes représente une relation de voisinage entre deux parcelles.

Un diagramme de Voronoï valide est nécessaire à la construction de ce graphe. Les seules données dont l'on a besoin en entrée sont donc les positions centrales des parcelles. Ces coordonnées peuvent être extraites de fichiers d'informations géographiques mais dans un premier temps, et dans un but illustratif, on se cantonne à utiliser des positions aléatoirement choisies.

#### IMAGE

La librairie Java JTS est spécialisée dans les traitements géométriques et permet notamment de générer un diagramme de Voronoï à partir d'une liste de points. Le résultat d'un tel traitement prend la forme d'un ensemble de polygones, chacun représentant une cellule de Voronoï, mais aucune autre information (d'adjacence par exemple) n'est fournie. Une fois le graphe parcellaire peuplé par des nœuds positionnés aux coordonnées fournies plus tôt, il reste donc à calculer les relations de voisinage.

#### IMAGE

Puisque l'on ne dispose que des polygones formant le diagramme de Voronoï et des positions de leur centroïdes, on procède par l'utilisation de tests géométriques. Naturellement, si deux cellules sont en contact – si elle partagent une arête – alors on relie les deux nœuds du graphe leur étant associés.

#### IMAGE

On observe sur la figure??? que les parcelles situées en bordure de la ville sont dotées

d’immenses cellules de Voronoï de par leur position. Afin de ne pas influencer négativement le déroulement de la simulation, dans laquelle l’aire des cellules peut avoir un impact sur l’évolution du système, on préfère modifier les contours du diagramme pour qu’il épouse la forme générale de la ville. Pour ce faire, on calcule l’enveloppe convexe du jeu de points utilisés comme coordonnées initiales, on l’agrandit de quelques unités afin que les parcelles ne soient pas collées à la bordure extérieure et on conserve uniquement l’intersection du diagramme original et de cette enveloppe.

IMAGE

DYNAMIQUE ?

### 3.2.2 Le graphe viaire

Le graphe viaire correspond à la couche routière. Ses nœuds sont des carrefours, aux croisements des parcelles, et ses arête des voies.

Comme lors de la construction du graphe parcellaire, on commence par placer les nœuds (ici des carrefours) puis on les relie par des arêtes (les routes).

Dans un premier lieu, il nous faut déterminer les positions des croisements qui feront office de nœuds dans le graphe viaire. Cette phase est basée sur l’analyse du graphe parcellaire construit plus tôt. Puisqu’une route entre deux bâtiments peut être définie par une arête de Voronoï partagée entre deux parcelles, il est naturel d’admettre qu’un croisement correspond à un

sommet de Voronoï partagée par deux cellules ou plus (voir figure ). Le but de l’étape décrite est donc de déterminer des groupes de cellules axées autour d’un même sommet pivot.

IMAGE

On remarque que ces clusters de parcelles semblent être des sous-graphes complets maximaux. Dans un premier temps, l’usage de l’algorithme de TRUC a été envisagée (et mis inutilement en place!) mais certains cas particuliers nous laisse entrevoir le fait qu’une contrainte supplémentaire est manquante. En effet, la recherche d’une clique maximale par l’algorithme cité ne prend pas en compte la nécessité que les cellules du cluster partagent un sommet. Sur l’exemple de la figure ???, quatres cellules forment une clique sans pour autant avoir un sommet de Voronoï en commun.

IMAGE TRIANGLE

ALGO PERSO

Les résultats sont enregistrés dans une structure de données associant à chaque carrefour sa position dans l’espace et les parcelles l’entourant mais le graphe viaire n’a pour l’instant pas été peuplé. Pour ce faire, on ajoute pour chaque croisement un nœud et comme pour le graphe parcellaire il faut encore connecter ces nœuds.

BLABLA DELAUNAY/VORONOI

## 4 Évolution

???

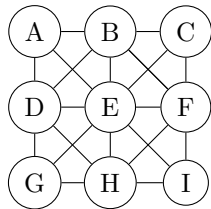
## Références

- [1] Niloy GANGULY et al. “A Survey on Cellular Automata”. Dans : (), p. 1–30.
- [2] U FRISCH, B HASSLACHER et U POMEAU. “Lattice-gas automata for the Navier-Stokes equation”. Dans : *Physical review letters* (1986).
- [3] A.R. KANSAL et al. “Cellular automaton of idealized brain tumor growth dynamics”. Dans : *BioSystems* 55.1-3 (fév. 2000), p. 119–127.
- [4] S.C. FU et George MILNE. “Epidemic modelling using cellular automata”. Dans : *Proc. of the Australian Conference on Artificial Life*. 2003.
- [5] Waldo TOBLER. “Cellular Geography”. Dans : (1975).
- [6] Anthony Gar-On YEH et Xia LI. “Urban Simulation Using Neural Networks and Cellular Automata for Land Use Planning”. Dans : *Advances in spatial data handling* (2002).
- [7] Paul M TORRENS et David O’SULLIVAN. “Cellular automata and urban simulation : where do we go from here?” Dans : *Environment and Planning B : Planning and Design* 28.2 (2001), p. 163–168.

- [8] Roger WHITE. “Cities and Cellular automata”. Dans : *Discrete Dynamics in Nature and Society* 2.2 (1998), p. 111–125.
- [9] Lucien BENGUIGUI, Daniel CZALANSKI et Rafael ROTH. *Modeling Cities in 3D : A Cellular Automaton Approach*. Rap. tech. 11. Israel Institute of Technology, 2004, p. 1829–1841.
- [10] Jeremy CORNU et Adrien DELCOURT. *Simulation de la dynamique spatiale urbaine par automate cellulaires*. Rap. tech.
- [11] Tom LECHNER, Seth TISUE et Andy MODDRELL. “Procedural Modeling of Urban Land Use”. Dans : ().
- [12] Edwige; DUBOS-PAILLARD, Yves; GUERMOND et Patrice LANGLOIS. “Analyse de l’évolution urbaine par automate cellulaire : le modèle SpaCelle”. Dans : *L’Espace géographique* 32 (2003), p. 357–.
- [13] Ferdinando SEMBOLONI. “The dynamic of an urban cellular automata in a 3-D spatial pattern”. Dans : *XXI National Conference*. 1997. Palermo, 2000, p. 20–22.
- [14] David O’SULLIVAN. “Graph-based Cellular Automaton Models of Urban Spatial Processes”. Thèse de doct. University of London, 2000.
- [15] David O’SULLIVAN. “Exploring Spatial Process Dynamics Using Irregular Cellular Automaton Models”. Dans : 33.1 (2001).
- [16] Thomas LECHNER et al. “Procedural City Modeling”. Dans : ().
- [17] Thomas LECHNER et al. “Procedural modeling of land use in cities”. Dans : *Cities* (2004).
- [18] George KELLY et Hugh MCCABE. “Citygen : An interactive system for procedural city generation”. Dans : *Fifth International Conference on* (2007).
- [19] Yoav I H PARISH et Pascal MÜLLER. “Procedural Modeling of Cities”. Dans : *SIGGRAPH*. 2001.
- [20] Guoning CHEN et al. “Interactive procedural street modeling”. Dans : *ACM SIGGRAPH 2008 papers on - SIGGRAPH ’08* (2008), p. 1.
- [21] P-A QUELOZ, B CHOPARD et Luthi P O. “Cellular automata model of car traffic in a two-dimensional street network”. Dans : *Journal of Physics* 0.111 (1996).
- [22] Basil WEBER et al. “Interactive Geometric Simulation of 4D Cities”. Dans : *Computer Graphics Forum* 28.2 (2009), p. 481–492.

|   |   |   |
|---|---|---|
| A | B | C |
| D | E | F |
| G | H | I |

(a) Automate cellulaire simple



(b) Automate cellulaire graphe

FIGURE 1