



HAL
open science

Association de la génération procédurale et des algorithmes génétiques pour modéliser la croissance urbaine

Etienne Tack, Gilles Énée, Frédéric Flouvat

► **To cite this version:**

Etienne Tack, Gilles Énée, Frédéric Flouvat. Association de la génération procédurale et des algorithmes génétiques pour modéliser la croissance urbaine. Extraction et Gestion des Connaissances, EGC'2025, Jan 2025, Strasbourg, France. pp.63-74. hal-04932361

HAL Id: hal-04932361

<https://hal.science/hal-04932361v1>

Submitted on 6 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Association de la génération procédurale et des algorithmes génétiques pour modéliser la croissance urbaine

Etienne Tack*, Gilles Énée*, Frédéric Flouvat**,

*ISEA (EA 7484), Université de la Nouvelle-Calédonie, Nouvelle-Calédonie
etienne.tack@etudiant.unc.nc, gilles.enee@unc.nc

**LIS (UMR 7020), Aix Marseille Université, France
frederic.flouvat@univ-amu.fr

Résumé. Cet article propose une méthode d'optimisation des influences spatiales dans des modèles multi-agents, en associant génération procédurale et algorithme génétique. Appliquée à un modèle de croissance urbaine, cette méthode permet aux agents, représentant les habitants, de prendre des décisions basées sur leur environnement (par exemple, la proximité d'une route favorisant la construction). La génération procédurale aide à modéliser ces influences, mais la complexité du réglage manuel des paramètres rend nécessaire l'usage d'un algorithme génétique afin de les optimiser automatiquement. L'approche étudie trois mesures spatiales (distance de Chamfer, densité de noyau, et grille de densité) pour entraîner le modèle, et simuler la localisation de nouvelles habitations. Les résultats expérimentaux démontrent l'efficacité de cette méthode, sa généralité et l'importance des fonctions de fitness.

1 Introduction

Cet article aborde les défis de la modélisation du développement urbain, un enjeu majeur dans certains pays, car les villes connaissent une croissance importante ces dernières années. Dans un tel contexte, les systèmes multi-agents sont largement utilisés pour modéliser le comportement des habitants et l'impact de leur environnement, notamment en termes de construction des habitations. Toutefois, ces travaux se concentrent principalement sur des villes régies par des règles d'urbanisme très strictes, et donc principalement connues à l'avance. Dans un certain nombre de pays, le développement des villes n'est pas régi par de telles règles (du moins pas complètement). Il dépend de règles tacites, de pratiques traditionnelles et/ou de contraintes géographiques, qui ne sont pas normalisées, mais influencent directement la construction de la ville. Dans ces zones dites "informelles", les systèmes multi-agents existants restent limités en raison d'hypothèses fortes, mais aussi de la complexité et de la diversité des situations.

En effet, de nombreuses tentatives ont été faites pour modéliser la croissance urbaine via des systèmes multi-agents, avec une représentation en grille ou vectorielle de l'espace (Schelling, 1971; Barros, 2004; Zhang et al., 2010; Jokar Arsanjani et al., 2013; Schwarz et al., 2016; Picascia et Yorke-Smith, 2017; Agyemang et al., 2022). Les modèles en grille sont plus simples à implémenter, mais sujets à des approximations spatiales. Par exemple, Schelling (1971) utilise une grille dans laquelle chaque cellule représente une zone résidentielle (avec plusieurs

habitations). Les agents choisissent de rester ou déménager, modélisant ainsi la ségrégation, en fonction de critères empiriques. Picascia et Yorke-Smith (2017) simule l'impact d'une migration de réfugiés dans une ville méditerranéenne, où les agents sélectionnent leur résidence en fonction du loyer et des politiques publiques, sans modéliser explicitement la localisation des bâtiments. Agyemang et al. (2022) étudie la croissance urbaine informelle au Ghana avec des agents qui évaluent les cellules de la grille (zones résidentielles) selon des critères tels que le loyer ou la proximité des commodités, mais la calibration du modèle est faite avec une méthode brute force (c.-à-d. que toutes tester toutes les possibilités sont testées sans stratégie de recherche). Augustijn-Beckers et al. (2011) propose une approche vectorielle plus précise, mais fondée sur des hypothèses spécifiques, comme des bâtiments de taille identique et un placement aléatoire le long des routes, limitant son applicabilité à des contextes plus variés.

En infographie, la génération procédurale est couramment utilisée pour créer des paysages et des villes virtuelles (Smelik et al., 2014). Parish et Müller (2001) présentent "CityEngine", qui modélise des villes en 3D, mais nécessite des connaissances expertes explicites et ne simule pas l'évolution urbaine. Emilien et al. (2012) utilisent des cartes d'intérêt pour générer des villages en fonction d'influences environnementales, mais la validation est uniquement visuelle et il manque les équations formelles des fonctions proposées.

Face aux limites des approches existantes, cet article propose une approche innovante, qui associe optimisation multi-critères et programmation procédurale, pour modéliser la position des bâtiments dans des zones formelles ou informelles. En s'appuyant sur les principes de la géographie de Tobler (1970), où les objets proches influencent davantage les nouvelles constructions, l'approche identifie deux types d'influences spatiales et les modélise par des fonctions mathématiques. Les paramètres de ces fonctions d'influences sont ensuite optimisés par un algorithme génétique (NSGA-II), qui s'appuie sur une fonction de récompense exploitant différentes informations spatiales.

La suite de l'article présente le modèle d'influence spatiale (section 2), explique comment les paramètres de ce modèle sont optimisés à l'aide d'un algorithme génétique (section 3) et décrit les expériences menées pour valider cette méthode (section 4).

2 Le sous-modèle d'influence

En général, les agents sont influencés par leur perception locale du monde. Dans notre travail, nous utilisons des fonctions d'influence pour déterminer où les agents peuvent construire de nouveaux bâtiments à chaque itération de la simulation. Une fonction d'influence est définie avec l'aide d'un expert (géographe) pour chaque facteur environnemental lié à la construction de nouveaux bâtiments.

Par exemple, la fonction d'attraction-répulsion dans la Figure 1 (premier graphique) illustre l'influence des bâtiments existants sur la construction d'un nouveau bâtiment. Dans cette figure, x représente la distance entre une position candidate pour un nouveau bâtiment et le bâtiment existant le plus proche, et y son influence sur la construction. En appliquant cette fonction sur une carte, nous obtenons une carte d'influence montrant l'attractivité de toutes les zones par rapport aux bâtiments existants (Figure 1.a). Les zones noires (positions des bâtiments existants) sont répulsives, tandis que les zones blanches sont attractives. Cela s'explique par le fait que les nouveaux bâtiments ne peuvent pas être construits les uns sur les autres, mais ils tendent à être proches les uns des autres (comme dans un village). La Figure 1.b) représente

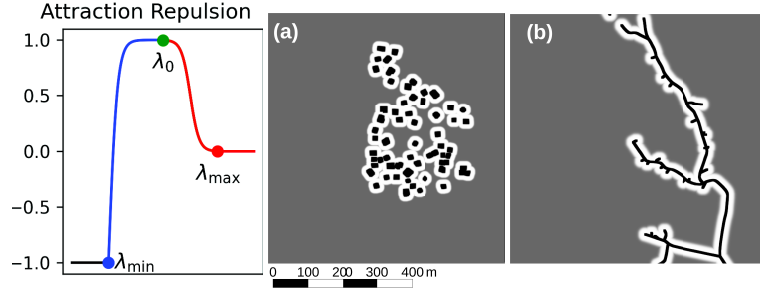


FIG. 1 – Fonction d’influence Attraction-Répulsion, et exemples d’application aux bâtiments (a) et routes (b) (Blanc = 1, Gris = 0 et Noir = -1).

les cartes d’influence des routes extrapolées à partir de cette même fonction d’influence. Sur la base de la Figure 1, la fonction d’attraction-répulsion peut être définie de manière plus formelle et plus générale comme suit :

$$AR(x) = \begin{cases} -1 & \text{if } x < \lambda_{\min} \\ f(x) & \text{if } \lambda_{\min} \leq x < \lambda_0 \\ g(x) & \text{if } \lambda_0 \leq x < \lambda_{\max} \\ 0 & \text{if } x \geq \lambda_{\max} \end{cases} \quad (1)$$

où λ_{\min} , λ_0 et λ_{\max} sont trois seuils de distance ($\lambda_{\min} < \lambda_0 < \lambda_{\max}$).

Dans l’équation 1, $f(x)$ représente l’augmentation de l’influence passant de répulsive à attractive, et $g(x)$ représente la diminution de l’influence passant d’attractive à neutre. Pour définir ces deux fonctions, nous utilisons la tangente hyperbolique (\tanh). Elle a été légèrement modifiée pour contrôler la pente de la courbe et la longueur de la transition entre une forte répulsion (c’est-à-dire -1) et une forte attraction (c’est-à-dire 1), comme base pour $f(x)$ et $g(x)$. Plus précisément, ces fonctions sont définies par :

$$\begin{aligned} f(x) &= \tanh \left((x - \lambda_{\min}) \times \frac{2\pi}{\lambda_0 - \lambda_{\min}} \right) * 2 - 1 \\ g(x) &= -\tanh \left(\left(x - \frac{\lambda_{\max} + \lambda_0}{2} \right) \times \frac{2\pi}{\lambda_{\max} - \lambda_0} \right) * \frac{1}{2} + \frac{1}{2} \end{aligned} \quad (2)$$

Différents types de facteurs ont des influences différentes, c’est-à-dire que plusieurs fonctions d’influence peuvent être identifiées par un expert et utilisées dans notre approche. Par exemple, la Figure 2 illustre les influences des chemins et de la pente telles que définies par notre expert. Cette influence est basée sur la fonction de distance ouverte illustrée dans la Figure 2. La fonction de distance ouverte peut également être définie en utilisant la même approche. Sa formule est donnée ci-dessous :

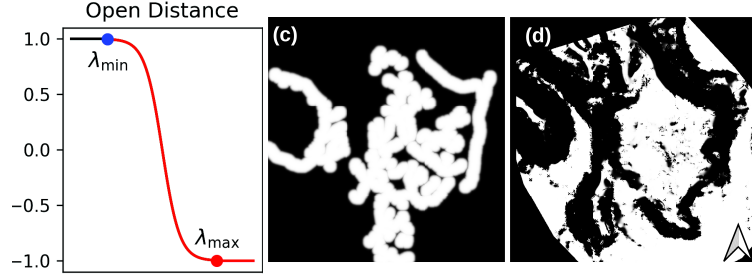


FIG. 2 – Fonction d'influence de distance ouverte et son application pour générer les cartes d'influence des chemins (c) et de la pente (d). (Blanc = 1, Gris = 0 et Noir = -1).

$$OD(x) = \begin{cases} 1 & \text{si } x < \lambda_{\min} \\ h(x) & \text{si } \lambda_{\min} \leq x < \lambda_{\max} \\ -1 & \text{si } x \geq \lambda_{\max} \end{cases} \quad (3)$$

$$h(x) = -\tanh\left(\left(x - \frac{\lambda_{\max} + \lambda_{\min}}{2}\right) \times \frac{2\pi}{\lambda_{\max} - \lambda_{\min}}\right)$$

Une fois que toutes les cartes d'influence sont traitées, elles sont pondérées et agrégées en une seule carte d'influence en utilisant la même équation introduite par Emilien et al. (2012).

$$\mathcal{I}(P) = \begin{cases} -1 & \text{si } \exists k \text{ tel que } I_k(P) = -1 \\ \max(0, \sum w_k \cdot I_k(P)) & \text{dans les autres cas.} \end{cases} \quad (4)$$

où P est une position candidate, $I_k(P)$ est la valeur d'attraction de la position P selon le facteur k dans la carte d'influence correspondante, et w_k est le poids pour chaque facteur d'influence k .

La Figure 3 montre les résultats de l'agrégation des quatre cartes d'influence présentées dans les Figures 1 et 2 (rendu à une résolution de 1 m). Contrairement aux travaux existants, les paramètres λ_{\min} , λ_0 et λ_{\max} de ces fonctions d'influence, ainsi que leur poids w_k , sont définis à l'aide d'un algorithme génétique comme détaillé dans la section suivante. Ces cartes d'influence sont ensuite utilisées lors de la simulation multi-agents pour définir l'emplacement des nouveaux bâtiments. Plus précisément, lorsqu'un agent cherche à déterminer où il va construire, il effectue une descente de gradient sur les valeurs de cette carte d'influence agrégée.

3 Optimisation des paramètres des fonctions d'influence à l'aide d'un algorithme génétique

Comme expliqué précédemment, le sous-modèle d'influence est défini par un ensemble de fonctions d'influence, leurs paramètres (λ_{\min} , λ_0 et λ_{\max} dans les Équations 1-3) et leurs

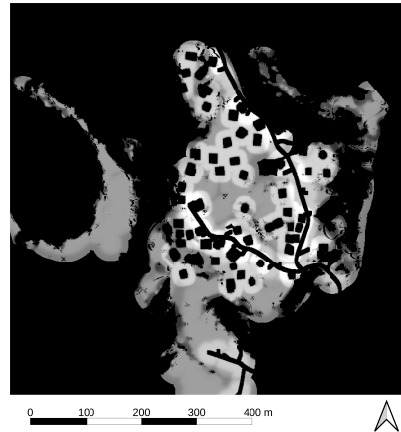


FIG. 3 – Carte des influences agrégées

poids pour la fonction d’agrégation (c’est-à-dire les paramètres w_k dans l’équation 4). Pour apprendre ces paramètres, l’algorithme génétique NSGA-II (Deb et al., 2002) a été choisi. NSGA-II signifie “Non-dominated Sorting Genetic Algorithm II” et il est largement utilisé comme un algorithme d’optimisation multi-objectifs. Par exemple, l’outil OpenMOLE l’utilise pour optimiser les paramètres de modèles multi-agents (Reuillon et al., 2013).

Les algorithmes génétiques nécessitent des fonctions de fitness pour évaluer la précision des solutions candidates, ici les paramètres et poids des fonctions d’influence. Les mesures comparent les données générées par le modèle multi-agents aux données observées. Le modèle est ensuite ajusté, exécuté et réévalué jusqu’à obtenir une erreur suffisamment faible. Dans ce contexte, la définition des fonctions de fitness est primordiale. Elles doivent notamment permettre de capturer finement la distribution spatiale des habitations. Dans notre travail, trois mesures ont été plus particulièrement étudiées.

La première (notée D dans les figures) mesure la différence de densité de bâtiments en s’appuyant sur un découpage de l’espace sous forme de grille. Comme illustré dans la Figure 4, l’espace est divisé en cellules, chacune contenant un certain nombre de bâtiments déterminé par leurs centroïdes. On en dérive ensuite une matrice de densité pour chaque jeu de données. La différence de densité est ensuite calculée comme la différence absolue entre les deux matrices.

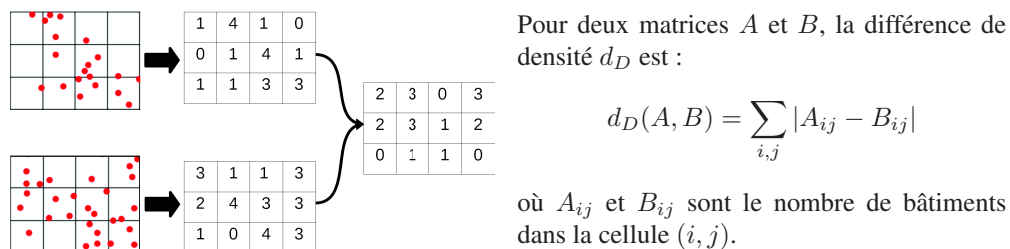
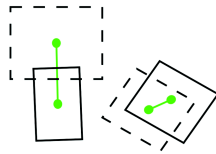


FIG. 4 – La différence de densité de bâtiments basée sur une grille.

La deuxième (notée C dans les figures) mesure la distance de Chamfer entre deux nuages de points. Elle somme les distances carrées de chaque point à son voisin le plus proche dans les données de référence (voir Figure 5). Cette mesure sera appliquée aux centroïdes des bâtiments et fournira plus d'informations sur les différences locales.

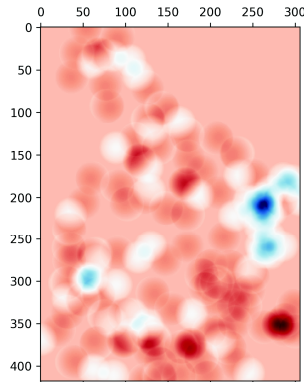


Pour deux sous-ensembles non vides X et Y , la distance de Chamfer $d_C(X, Y)$ est définie par :

$$d_C(X, Y) = \sum_{x \in X} \min_{y \in Y} \|x - y\|^2 + \sum_{y \in Y} \min_{x \in X} \|x - y\|^2$$

FIG. 5 – La distance de Chamfer entre les centroïdes des bâtiments.

La dernière (notée K dans les figures) mesure de validation étudiée est la différence entre deux estimations de densité de noyau (*kernel density*) de Parzen (1962). Cette densité est souvent considérée comme plus représentative que la simple densité ponctuelle. Elle lisse la densité en tenant compte non seulement des points individuels, mais aussi de l'influence des points environnants, ce qui permet une représentation plus continue et réaliste des phénomènes spatiaux. Nous appliquons cette méthode pour générer deux cartes de chaleur (une pour les données simulées et une pour les données réelles), puis nous calculons la différence entre ces deux cartes et prenons la moyenne des différences absolues pour chaque pixel. La Figure 6 montre la différence de densité de noyau entre deux cartes et détaille comment cette information est agrégée en une valeur.



$$d_K(A, B) = \frac{1}{wh} \sum_{i=1}^w \sum_{j=1}^h \left| \hat{f}_A(i, j) - \hat{f}_B(i, j) \right|$$

où :

- \hat{f}_A est la densité de noyau sur les centroïdes des bâtiments générés par la simulation,
- \hat{f}_B est la même chose, mais pour les données de référence,
- (w, h) est la taille du rectangle qui englobe l'ensemble des données combiné.

FIG. 6 – La différence de densité de bâtiments basée sur une estimation par noyau.

4 Expériences

4.1 Jeux de données

Dans un premier temps, nous étudierons le quartier informel de Valencina, près de Suva aux Fidji. Ce quartier mesurant 118885 m^2 a été observé sur 4 dates (1994, 2002, 2009 et

2019), et pour chacune les bâtiments ont été identifiés (61 en 1994, 83 en 2002, 103 en 2009, et 129 en 2019). Pour ces dates, les routes et chemins ont aussi été extraits des images. Un modèle numérique de terrain (MNT) généré en 2019 a aussi été utilisé pour extraire la pente.

Dans un second temps, nous ferons des expérimentations sur 7 autres territoires (pas uniquement informels) répartis partout dans le monde (cf. Fig. 11). Ces données sont issues du défi SpaceNet 7 (Etten et al., 2021), et contiennent l’emprise des bâtiments. Les données du réseau routier ont été extraites de (OpenStreetMap contributors, 2017). La pente de ces quartiers a été calculée à partir des modèles numériques de terrains de FABDEM (Hawker et al., 2022).

4.2 Paramétrage de l’algorithme génétique

Notre code, développé en Python, est disponible sur GitHub (https://github.com/etienne-tack/spatial_influences). Le modèle d’agents pour les habitats informels est construit avec MESA, et nous avons utilisé PyMOO (Blank et Deb, 2020) pour l’optimisation des paramètres des fonctions d’influence. Un expert a associé des fonctions d’attraction-répulsion pour représenter l’influence des bâtiments et des routes, et des fonctions de distance ouverte pour celles des chemins et de la pente.

Les paramètres par défaut de NSGA-II dans PyMOO ont été utilisés : mutation polynomiale (indice de perturbation 20) et croisement binaire simulé (*simulated binary crossover*, avec un indice de distribution à 15 et une probabilité de 0.9). La taille de la population est fixée à 50 individus et l’algorithme s’arrête après 300 générations ou si les valeurs des objectifs sont stables pendant 35 générations.

Pour identifier la fonction de fitness optimale pour NSGA-II, sept combinaisons de mesures de validation ont été testées sur trois périodes du jeu de données Valencina (1994-2002, 2002-2009, 2009-2019). La première différence de densité a été calculée sur la base d’une grille de $50 m \times 50 m$, et NSGA-II a été exécuté 50 fois pour chaque combinaison et période.

4.3 Évaluation des modèles d’influences

Pour analyser les résultats obtenus, nous avons commencé par comparer la carte d’influence générée à partir des fonctions obtenues à partir des paramètres optimisés par NSGA-II, avec la carte contenant la localisation des vrais bâtiments. Pour ce faire, nous avons mesuré combien de nouveaux bâtiments dans les données de terrain sont placés dans des zones marquées comme répulsives par le modèle (-1). Ces bâtiments mettent en évidence les erreurs commises par le sous-modèle d’influence. Cette erreur du modèle est définie par la formule suivante :

$$\text{Erreur d'influence} = \frac{\text{Nombre de nouveaux bâtiments dans les zones répulsives}}{\text{Nombre de nouveaux bâtiments}}$$

La Figure 7 montre l’erreur d’influence moyenne pour chaque combinaison de mesures de validation (c’est-à-dire chaque fonction de fitness dans NSGA-II). Les barres verticales (rouges) représentent l’erreur obtenue pour des paramètres aléatoires. Les pourcentages indiquent la différence entre l’erreur moyenne obtenue par notre approche et celle obtenue avec des paramètres aléatoires. Cette figure montre que NSGA-II a pu optimiser les paramètres des fonctions d’influence spatiales grâce aux mesures de validation choisies. En moyenne, les valeurs les plus basses d’erreur d’influence sont atteintes en utilisant la mesure de différence de densité basée sur la grille (notée D) combinée avec au moins une autre mesure.

Génération procédurale et algorithmes génétiques pour modéliser la croissance urbaine

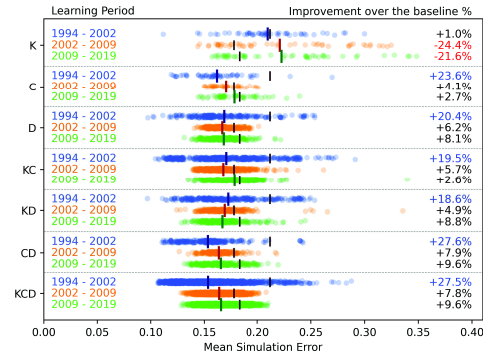
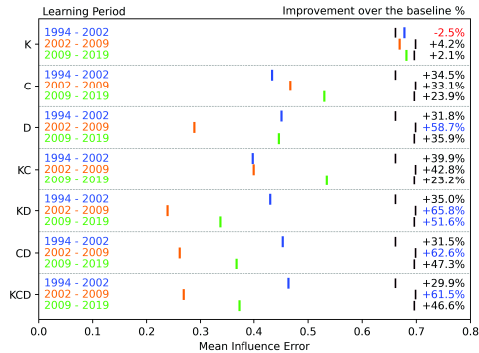


FIG. 7 – Erreur d'influence pour chaque combinaison de mesures et pour chaque période (Valencina).

FIG. 8 – Erreurs de simulation pour chaque combinaison de mesures et chaque période (Valencina).

4.4 Évaluation des simulations

Bien que l'erreur d'influence fournisse des informations sur la qualité des fonctions, elle ne garantit pas le réalisme des simulations. Une nouvelle mesure d'erreur, basée sur les résultats de simulation, est donc nécessaire. Une distance euclidienne normalisée a été utilisée pour comparer les nouveaux bâtiments simulés avec ceux observés en réalité. Lors du calcul, chaque bâtiment de validation sélectionné est retiré de l'ensemble de validation, évitant toute redondance.

La Figure 8 compare les résultats de simulation obtenus avec les modèles d'influence issus des différentes combinaisons de fonctions de fitness, avec ceux obtenus avec une simulation aléatoire (en rouge). Les meilleurs résultats sont encore obtenus avec les modèles entraînés avec la différence de densité basée sur une grille, combinée avec au moins une autre mesure.

4.5 Comparaison des modèles d'influence optimisés

La Figure 9 montre les fonctions d'influence obtenues avec la meilleure combinaison de mesures (CD). Bien que les solutions soient similaires pour les périodes 1994-2002 et 2009-2019, il n'existe pas de modèle unique valable pour 1994 à 2019 et les paramètres d'influence obtenus pour 2002 sont très différents des autres.

L'influence des bâtiments et des routes sur les constructions est faible en 2002 (autour de zéro). L'influence des chemins semble plus importante. La pente en 2002 n'a une influence négative que lorsqu'elle est proche de 2 radians (axe des x), alors que cette valeur est bien plus basse pour 1994 et 2009 (0,5 radian).

Ces résultats sont encourageants, car l'approche tente de simuler les choix humains en ne tenant compte que des facteurs environnementaux, bien que d'autres facteurs, comme les aspects socio-économiques, influencent leur comportement.

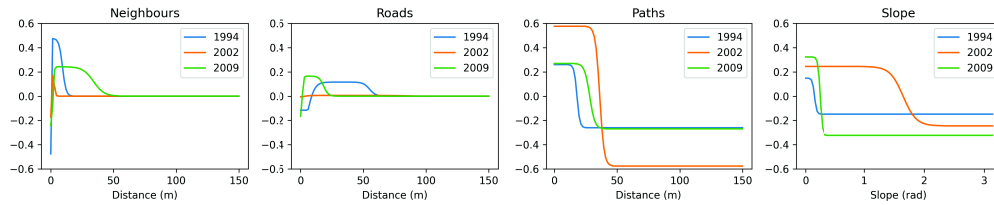


FIG. 9 – Fonctions d’influence pour les meilleures solutions obtenues avec la distance de Chamfer et les mesures de densité basées sur une grille (Valenicina).

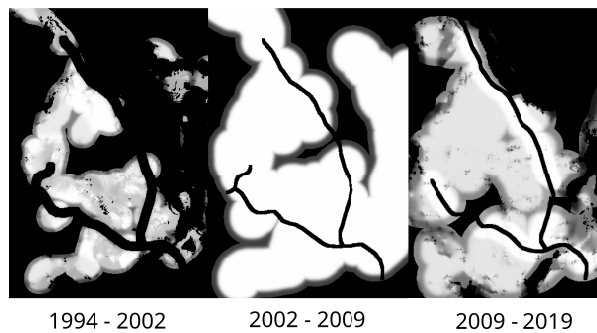


FIG. 10 – Cartes d’influence des fonctions d’influence affichées dans la Figure 9.

4.6 Expérimentations sur des données plus larges et hétérogènes

Pour évaluer la généricité de notre méthode sur la modélisation de la croissance urbaine, nous avons mené une série d’expériences avec les données SpaceNet 7 (Etten et al., 2021). Ces données, plus diversifiées, incluent des villes aux densités urbaines et caractéristiques environnementales variées, ce qui est idéal pour tester la généralisation de notre modèle à différentes morphologies urbaines.

Le paramétrage de l’algorithme génétique est le même que pour les expérimentations précédentes. Comme pour Valenicina, chaque jeu de données a été entraîné séparément. La seule différence est l’utilisation d’une seule fonction d’influence pour les routes et les chemins, car il n’y a pas de distinctions entre routes et chemins dans les données en entrée.

La Figure 13 (à gauche) montre l’erreur d’influence pour chaque solution obtenue. Les résultats sont similaires à ceux de Valenicina, et l’erreur de simulation (Figure 13, à droite) est plus basse. L’erreur d’influence est particulièrement sensible à la qualité des données de terrain, par exemple si le réseau routier n’est pas spatialement précis, comme c’est le cas pour Londres ou le Caire 2 par exemple, ce qui affecte les performances.

La Figure 12 illustre la relation entre le temps d’exécution par génération et le nombre de nouveaux bâtiments. Naturellement, plus il y a de bâtiments, plus le temps de simulation augmente, mais les temps restent acceptables dans l’ensemble. Par exemple, les simulations sur les données de Tripoli et du Caire 1, ont un temps d’exécution moyen autour de 200 secondes, tandis que New York, Londres et Santiago, avec moins de bâtiments, prennent moins de 50

Code donné par SpaceNet 7 Nom simplifié	Taille (km ²)	Delta du nombre de bâtiments
L15-0368E-1245N_1474_3210_13 Los Angeles	15.58	366
L15-0577E-1243N_2309_3217_13 New York	15.83	169
L15-0632E-0892N_2528_4620_13 Santiago	19.58	122
L15-1025E-1366N_4102_2726_13 Londre	9.37	326
L15-1138E-1216N_4553_3325_13 Tripoli	17.25	1001
L15-1203E-1203N_4815_3378_13 Caire 1	17.96	1002
L15-1204E-1202N_4816_3380_13 Caire 2	17.24	240

FIG. 11 – Données SpaceNet 7.

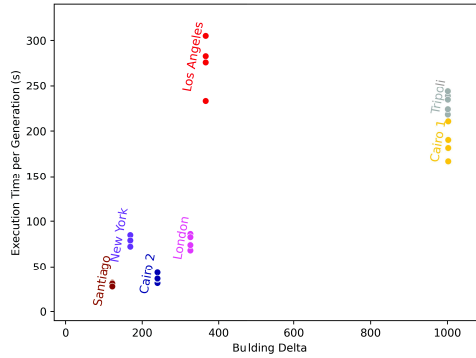


FIG. 12 – Temps d'exécution en relation avec la taille du delta des bâtiments pour chaque jeu de données.

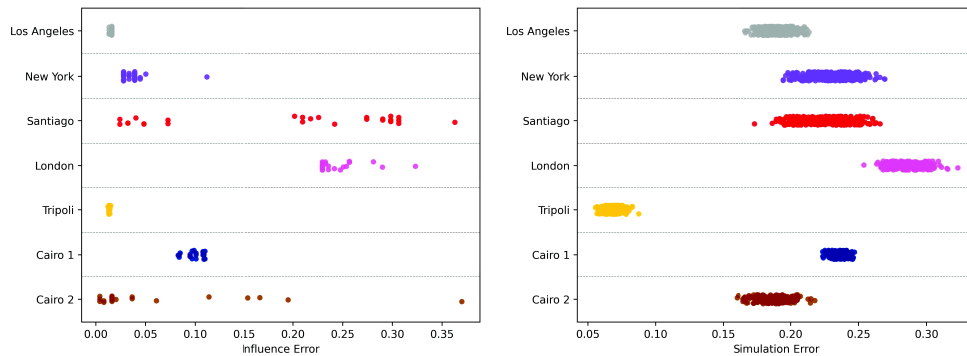


FIG. 13 – Erreur d'influence et erreur de simulation pour les données SpaceNet 7.

secondes.

5 Conclusions et Perspectives

L'approche proposée associe génération procédurale et algorithmes génétiques pour modéliser les influences spatiales des facteurs environnementaux (bâtiments, routes, pente, etc.) dans une simulation multi-agents. Deux types d'influences identifiées par un expert (attraction-répulsion et distance ouverte) sont représentées par des fonctions mathématiques, dont les paramètres et les poids sont déterminés automatiquement par l'algorithme génétique NGSA-II.

Trois mesures de validation spatiale ont été combinées comme fonctions de fitness pour cet algorithme génétique : la différence de densité basée sur une grille, la distance de Chamfer et la différence de densité de noyau. Les résultats montrent que la différence de densité basée sur un découpage de l'espace en grille offre les meilleures performances, bien que sa précision

puisse paraître inférieure intuitivement. Une légère amélioration est obtenue en la combinant avec la distance de Chamfer.

Cependant, des écarts notables entre les résultats de la simulation et la répartition réelle des bâtiments persistent. Selon les experts, cela s’explique probablement par l’absence du facteur humain dans le modèle, un élément pourtant reconnu dans la littérature pour leur impact majeur sur la croissance urbaine.

Il existe plusieurs perspectives à ce travail. Une première perspective est d’utiliser un algorithme génétique permettant de considérer l’ensemble de la série temporelle comme entrée (et non seulement chaque période de temps indépendamment). Des travaux récents ont abordé ce problème dynamique d’optimisation multi-objectifs Zou et al. (2021), et il serait intéressant d’étudier leur efficacité dans notre contexte. Indépendamment des emplacements des bâtiments, nous travaillons actuellement sur l’apprentissage de formes réalistes pour les bâtiments. Dans le modèle, il serait aussi intéressant d’ajouter la génération automatique de routes et de chemins à notre modèle (comme dans Emilien et al. (2012)) afin de prendre en compte également leur évolution. Enfin, notre modèle multi-agents doit également être complété pour intégrer le comportement humain (règles de comportement des agents). Comme mentionné précédemment, la construction de bâtiments n’est pas seulement guidée par des facteurs environnementaux. Elle est également étroitement liée à des facteurs socio-économiques et politiques. Un projet est en cours avec des historiens, géographes, économistes et anthropologues pour intégrer leur connaissance à ce sujet.

Références

- Agyemang, F. S. K., E. Silva, et S. Fox (2022). Modelling and simulating ‘informal urbanization’ : An integrated agent-based and cellular automata model of urban residential growth in Ghana. *Environment and Planning B : Urban Analytics and City Science* 50(4), 863–877.
- Augustijn-Beckers, E.-W., J. Flacke, et B. Retsios (2011). Simulating informal settlement growth in Dar es Salaam, Tanzania : An agent-based housing model. *Computers, Environment and Urban Systems* 35(2), 93–103.
- Barros, J. (2004). Simulating Urban Dynamics in Latin American Cities. In P. Atkinson, F. Wu, G. Foody, et S. Darby (Eds.), *GeoDynamics*, pp. 313–328. CRC Press.
- Blank, J. et K. Deb (2020). pymoo : Multi-objective optimization in python. *IEEE Access* 8, 89497–89509.
- Deb, K., A. Pratap, S. Agarwal, et T. Meyarivan (2002). A fast and elitist multiobjective genetic algorithm : Nsga-ii. *IEEE transactions on evolutionary computation* 6(2), 182–197.
- Emilien, A., A. Bernhardt, A. Peytavie, M.-P. Cani, et E. Galin (2012). Procedural generation of villages on arbitrary terrains. *The Visual Computer* 28(6-8), 809–818.
- Etten, A. V., D. Hogan, J. Martinez-Manso, J. Shermeyer, N. Weir, et R. Lewis (2021). The multi-temporal urban development spacenet dataset.
- Hawker, L., P. Uhe, L. Paulo, J. Sosa, J. Savage, C. Sampson, et J. Neal (2022). A 30 m global map of elevation with forests and buildings removed. *Environmental Research Letters* 17(2), 024016.

- Jokar Arsanjani, J., M. Helbich, et E. de Noronha Vaz (2013). Spatiotemporal simulation of urban growth patterns using agent-based modeling : The case of Tehran. *Cities* 32, 33–42.
- OpenStreetMap contributors (2017). Planet dump retrieved from <https://planet.osm.org> . <https://www.openstreetmap.org>.
- Parish, Y. I. et P. Müller (2001). Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 301–308.
- Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics* 33(3), 1065–1076.
- Picascia, S. et N. Yorke-Smith (2017). Towards an agent-based simulation of housing in urban beirut. In *Agent Based Modelling of Urban Systems*, pp. 3–20. Springer International Publishing.
- Reuillon, R., M. Leclaire, et S. Rey-Coyrehourcq (2013). OpenMOLE, a workflow engine specifically tailored for the distributed exploration of simulation models. *Future Generation Computer Systems* 29(8), 1981–1990.
- Schelling, T. C. (1971). Dynamic models of segregation. *Journal of mathematical sociology* 1(2), 143–186.
- Schwarz, N., J. Flacke, et R. Sliuzas (2016). Modelling the impacts of urban upgrading on population dynamics. *Environmental Modelling & Software* 78, 150–162.
- Smelik, R. M., T. Tutenel, R. Bidarra, et B. Benes (2014). A survey on procedural modelling for virtual worlds. In *Computer Graphics Forum*, Volume 33, pp. 31–50. Wiley Online Library.
- Tobler, W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic geography* 46(sup1), 234–240.
- Zhang, H., Y. Zeng, L. Bian, et X. Yu (2010). Modelling urban expansion using a multi agent-based model in the city of changsha. *Journal of Geographical Sciences* 20, 540–556.
- Zou, F., G. G. Yen, L. Tang, et C. Wang (2021). A reinforcement learning approach for dynamic multi-objective optimization. *Information Sciences* 546, 815–834.

Summary

This paper proposes a method for modelling and optimising spatial influences in agent-based models by combining procedural generation with a genetic algorithm. Applied to an urban growth model, this method enables agents, representing residents, to make decisions based on their environment (e.g., proximity to roads encouraging construction). Procedural generation aids in modeling these influences, but the complexity of manually adjusting the parameters necessitates the use of a genetic algorithm for automatic optimization. The approach examines three spatial measures—Chamfer distance, kernel density, and density grid—to train the model and simulate the location of new housing. Experimental results demonstrate the method’s effectiveness, generality, and the critical role of fitness functions.