

Recherche locale et théorie des jeux appliquées à la création de typo-morphologies compactes

Compact typo-morphologies by use of local search and game theory methods

Quentin Meurisse^{1,*}, Isabelle De Smet², Hadrien Mélot¹, David Laplume², Thomas Brihaye¹, Cédric Rivière¹, Emeline Coszach², Jérémy Cenci², Sésil Koutra² et Vincent Becue²

¹UMONS, Faculté des Sciences, 7000 Mons, Belgique

²UMONS, Faculté d'Architecture et d'Urbanisme, 7000 Mons, Belgique

Résumé. En vue d'une densification urbaine durable, un outil ayant pour but d'évaluer et d'assister la conception d'îlots urbains compacts pourvus d'une densité de population cible a été créé et testé dans le cadre du projet CoMod. Le concept de compacité spatiale est appliqué ici, à l'échelle architecturale, sur le bâti, le non-bâti et les deux combinés. Cette approche encourage les typo-morphologies économes en terrain et en ressources matérielles tout en étant efficaces d'un point de vue énergétique. Afin d'éviter une potentielle exagération de ce concept divers critères notamment relatifs aux espaces verts, aux ombres ainsi que des distances et surfaces minimales sont considérés. Cependant, viser la compacité urbaine rencontre une conciliation difficile entre les divers critères quantitatifs et qualitatifs. De nombreux outils mathématiques ont déjà été appliqués à des problèmes urbanistiques (méthodes d'optimisation, aide à la décision, automates cellulaires, ensembles fractals, etc.). L'étude de typo-morphologies compactes avec l'aide de la théorie des jeux ou de la recherche locale peut aider à la gestion des problèmes provenant de critères conflictuels. Dans cet article nous présentons un prototype de programme qui génère des îlots urbains en utilisant la recherche locale et la théorie des jeux.

Mots-clés. Scan2020, Processus et méthodologie, Îlot urbain, Recherche locale, Théorie des jeux

Abstract. In prospect of sustainable urban densification, a tool aiming to assess and to assist the design of compact housing blocks with a target population density was created and tested in the scope of the CoMod Project. The concept of spatial compactness is here applied, at the architectural scale, on the built environment, the non-built environment and both combined. This approach encourages typo-morphologies which save land and material resources while achieving high energy efficiency. Potential misuse of the

* Auteur correspondant : quentin.meurisse@umons.ac.be

concept is prevented by numerous objective criteria notably relative to green areas, projected shadows as well as minimal distances and surfaces to consider. However, targeting urban compactness faces a difficult conciliation between various quantitative and qualitative parameters. Numerous mathematical tools have already been applied on the problem of urban planning (using optimization methods, multi-criteria decision help, cellular automata, fractal sets, etc.). The study of compact typomorphologies with the help of game theory or local search can help solving the problems that arise when dealing with conflicting criteria. We are currently developing a prototype of software that generates urban blocks using local search algorithms and game theory methods.

Keywords. SCAN2020, Process and Methodology, Compactness, Urban block, Local search, Game theory.

1. Introduction

Dans le cadre du projet CoMod, un outil ayant pour but d'évaluer et d'assister la conception d'îlots urbains compacts pourvus d'une densité de population cible a été créé et testé, et ce en vue d'une densification urbaine durable (De Smet, 2018).

Reconnue pour sa relation avec les formes, la compacité spatiale, via ses indicateurs volumiques, surfaciques ou de porosité, peut permettre de réduire le foncier et les superficies d'enveloppe du bâti tout en exploitant les qualités spatiales des espaces de forme compacte. Cette approche encourage les typo-morphologies économes en terrain et en ressources matérielles tout en étant efficaces d'un point de vue énergétique. Appliquer à l'aide de l'outil le concept de compacité spatiale permet de renforcer les liens entre densité et typomorphologies (De Smet, 2018).

Divers auteurs en développent le concept à l'échelle de la ville, en complément à la densité (Bertaud & Malpezzi, 2003 ; Bonin & Tomasoni, 2013 ; Dantzig & Saaty, 1973 ; Frankhauser, 2002 ; Frankhauser, Tannier, Vuidel, & Houot, 2008 ; Halleux J.-M., 2012 ; Jenks, Burton, & Williams, 1996) (Katz, 1993 ; Kirwan, 1992 ; Maignant, 2005 ; Pouyane, 2004). D'autres l'appliquent à l'échelle du bâti pour les économies d'énergie (Pupille J.-M., 2013 ; Arantes, Marry, Baverel, & Quenard, 2016 ; Grenier, 2007) et de matériaux (Marique, 2013). Les méthodes de calcul de la compacité surfacique utilisées couramment à l'échelle macro et celles de la compacité du volume bâti sont exploitées dans l'outil à l'échelle infra-locale et plus précisément à l'échelle de l'îlot afin d'en étudier le bâti, le non-bâti et l'ensemble résultant. Réduire les différentes valeurs de compacité afin de répondre au critère de compacité spatiale est un des objectifs de l'outil (De Smet, 2018).

Néanmoins, une attention particulière est portée sur les répercussions spatiales dérivées du processus. En effet, un ensemble de contraintes spatiales et morphologiques doit être pris en compte lors de la création d'îlots de logements compacts afin d'éviter une augmentation exagérée de la compacité tout en favorisant les qualités spatiales des espaces compacts. Un ensemble de contraintes ont été définies (De Smet & Laplume, 2019) et seront prises en considération dans notre étude :

- un volume et une superficie d'habitat minimums par type de logement ;
- un pourcentage de superficie d'espaces verts par îlot dont 50% sont constitués de maximum deux espaces de forme considérée comme compacte ;
- un apport de luminosité suffisant sur les façades, les espaces extérieurs et les espaces intérieurs ;

- une distance minimale entre façades du point de vue de la promiscuité et de l'accessibilité des services de secours ;
- un sentiment de fermeture de l'ensemble et de ses composants permettant notamment une certaine porosité entre l'intra-îlot et l'extérieur.

Cibler la compacité urbaine se heurte donc à une conciliation difficile entre divers paramètres quantitatifs et qualitatifs. Dans son état actuel, l'outil est un outil statique où concilier l'ensemble des paramètres nécessite une utilisation « par tâtonnement », rendue fastidieuse par un encodage manuel des paramètres. Ces paramètres doivent être préalablement calculés par l'utilisateur à chaque étape de la conception, et ce, en raison du caractère itératif du processus de conception. Une facilité considérable pourrait découler d'un interfaçage de l'outil avec un logiciel de représentation 3D. Cette disposition permettrait le calcul automatique de la majorité des indicateurs, sur base d'un modèle DAO de l'îlot, dans toutes ses composantes. Il deviendrait alors possible de guider plus efficacement l'utilisateur de l'outil, sur base d'une configuration donnée d'un projet, quant à l'intérêt d'une modification spécifique envisagée. Cette approche nécessiterait un procédé permettant de hiérarchiser les indicateurs et contraintes. L'outil aurait alors toutes les propriétés nécessaires à la mise en œuvre d'une véritable démarche d'optimisation, au sens mathématique du terme.

Dans cette optique, le Projet CoMod (Compacité urbaine sous l'angle de la modélisation mathématique) porté par une équipe pluridisciplinaire a été mis en place. Ce projet est supervisé par trois départements de recherche de l'Université de Mons (Belgique) : le Service Projets, Ville et Territoire (Faculté d'Architecture et d'Urbanisme), le Service d'Algorithmique et le Service de Mathématiques Effectives (Faculté des Sciences). De nombreux outils mathématiques ont déjà été appliqués sur des problèmes urbanistiques (notamment des méthodes d'optimisation, l'aide à la décision multicritères, les automates cellulaires, les ensembles fractals, *etc.*). Ces divers outils ont chacun leur utilité dans certains problèmes. Les modèles issus de la recherche locale et de la théorie des jeux nous semblent les plus à même d'aider à la résolution des problèmes d'optimisation et de gestions de critères conflictuels survenant lors de l'étude de typo-morphologies compactes. Nous présentons dans cet article un prototype de programme générant des configurations d'îlots urbains utilisant dans un premier temps des algorithmes de recherche locale et ensuite des méthodes issues de la théorie des jeux. Les îlots générés optimisent des critères de compacité tout en respectant des contraintes légales mais également des contraintes portant sur la vie privée et le confort de la population.

2. Méthodologie

2.1 Contexte urbanistique

La réflexion initiale de développement du logiciel suit la méthodologie de construction d'un catalogue de configurations, issu d'une expérimentation résolument théorique de l'outil "statique" (De Smet, 2018). Ce catalogue, composé au total de 1674 cas, résulte de la variation systématique de paramètres clés sur des îlots fictifs triangulaires, rectangulaires et carrés. La superficie des îlots considérés varie, ainsi que la profondeur de bâtisse (de 10 à 18m, par pas de 2m) et le nombre de niveaux (de 1 à 8). Limiter le modèle à 8 étages permet son implémentation dans plusieurs contextes urbains.

Quatre familles de configurations sont considérées dans ce catalogue :

- fermée (périmètre de l'îlot intégralement bâti) ;

- percée (présence de failles entre l'extérieur de l'îlot et l'intra-îlot) ;
- augmentée (ajout de volume bâti en intra-îlot) ;
- échelonnée (présence de hauteurs de bâti différentes).

Ces configurations sont issues des hypothèses suivantes :

- une densité élevée combinée à un gabarit restreint ne pourra être obtenue a priori qu'en ayant recours à une emprise au sol plus importante que pour des constructions plus élevées ;
- les impératifs d'éclairage, et donc de salubrité, limitent la profondeur de bâtisse ;
- pour une profondeur de bâtisse fixée, la typo-morphologie permettant la plus grande emprise au sol privilégie de facto la construction prioritairement en périmètre d'îlot ;
- une telle construction en périmètre d'îlot est conforme à la volonté majoritaire des gouvernances de favoriser les implantations en front de voirie (ou en alignement avec le bâti existant) ;
- une typo-morphologie partiellement fermée peut permettre la création et la détermination d'espaces extérieurs privés ou semi-privés intimisés vis-à-vis de l'espace public.

Ce catalogue a permis un ensemble d'observations relatives notamment à l'évolution des densités de population et des indicateurs de compacité selon les variables considérées, afin de fournir des pistes et des supports de réflexion dans le cadre de projets concrets. Néanmoins, les types de configurations sont limitées et compléter le catalogue nécessiterait un travail d'encodage et de diagnostic très rébarbatifs.

2.2 L'outil dynamique utilisant la recherche locale

L'outil « dynamique » en construction permettrait de dépasser les limites tant d'arbitrage que d'encodage et de développer une réelle optimisation typo-morphologique des îlots compacts. Il a pour but de fournir des configurations d'îlots urbains qui optimisent certains critères et contraintes. En plus de cela, l'outil « dynamique » est implémenté de sorte à laisser à l'utilisateur une certaine marge de manœuvre niveau choix des critères à prendre en compte ainsi que de leurs paramètres.

Pour réaliser cela, la première version du programme utilise des algorithmes de recherche locale (El-Ghazali, 2009; De Beukelaer, Davenport, De Meyer, & Fack, 2017). En informatique, la recherche locale est une méthode pour résoudre des problèmes d'optimisation. Ce genre de problème peut être formulé comme suit : étant un ensemble de solutions pour un problème, nous voulons trouver les solutions qui maximisent ou minimisent un certain critère appelé fonction objectif.

Par exemple, imaginons que nous voulions trouver le meilleur mot de passe composé de quatre chiffres. Toutes les combinaisons de quatre chiffres sont des solutions. Ainsi 0000 et 1234 sont des solutions, peut-être mauvaises mais ce sont des solutions quand même.

Pour comprendre comment fonctionne la recherche locale basique, nous avons besoin de la notion de *voisinage*. Étant donné une solution s , un voisinage de s est l'ensemble des solutions qui peuvent être atteintes en appliquant une petite transformation à s ; un élément de cet ensemble est appelé *un voisin* de s . Retournons à notre exemple du mot de passe à quatre chiffres et considérons que notre opération de voisinage consiste simplement à changer uniquement un chiffre de notre solution courante. Des voisins de 0000 sont donc 1000, 0200 ou 0007.

Une fois que nous avons défini l'ensemble de solutions, la fonction objectif et l'opération de voisinage, la recherche locale basique fonctionne comme suit : nous démarrons d'une solution quelconque, nous calculons son voisinage, nous évaluons chacun de ses voisins avec la fonction objectif, le meilleur voisin devient notre solution courante s'il est meilleur que cette dernière et nous itérons cette procédure. L'algorithme s'arrête s'il trouve une solution meilleure que tous ses voisins ou satisfait un critère d'arrêt (par exemple la recherche ne peut pas durer plus d'un certain temps).

En plus de trouver une solution optimisant la fonction objectif, la recherche locale peut aussi essayer de satisfaire certaines contraintes. Il existe deux types de contraintes : les contraintes obligatoires et les contraintes pénalisantes. Pour satisfaire une contrainte obligatoire, la recherche locale ignore simplement les solutions qui ne vérifient pas un certain critère. Les contraintes pénalisantes pénalisent les solutions qui ne respectent pas un critère donné. Ainsi, la recherche locale cherche maintenant à trouver la meilleure solution qui n'est pas pénalisée par les contraintes.

Une faiblesse de la recherche locale est qu'elle ne garantit pas que nous obtenions finalement la meilleure solution. En effet, la solution retournée est meilleure que tous ses voisins, pas nécessairement meilleure que toutes les autres solutions. C'est la meilleure *localement* et non *globalement*. Afin d'essayer d'obtenir la meilleure solution possible, nous avons besoin d'autres techniques de recherche appelées *métaheuristiques*. El-Ghazali (El-Ghazali, 2009) définit les métaheuristiques comme suit : "*Metaheuristic search methods can be defined as upper level general methodologies (templates) that can be used as guiding strategies [...] to solve specific optimization problems*". Ce que nous appelons "la recherche locale basique" est un métaheuristique, the Hill Climbing. Il existe des métaheuristiques qui acceptent des voisins de moins bonne qualité. Cela peut mener à de meilleures solutions que celles retournées par le Hill Climbing. Ces variantes pourraient être étudiées à l'avenir afin d'améliorer le programme.

2.3 Modélisation de notre problème

Expliquons maintenant comment nous utilisons la recherche locale afin de nous aider à remplir un îlot urbain, dans un premier temps uniquement avec des bâtiments. Nous insistons sur le fait que nous utilisons la recherche locale sur un problème spécifique. Néanmoins, la recherche locale est très générale et peut être utilisée sur un autre problème, avec un autre modèle ou des autres contraintes. Dans notre contexte, une solution (un îlot urbain) est modélisée par un carré ou un rectangle quadrillé. Chaque cellule peut avoir différentes couleurs. Une cellule blanche est une cellule non-bâtie. Une cellule avec une autre couleur ou un autre motif est bâtie. Un bloc de cellules avec le même motif est un bâtiment. Pour l'instant, les bâtiments n'ont pas de hauteur. Nous nous intéressons à leur surface sur l'îlot.

Ce que nous venons de présenter est la base de notre modélisation d'un îlot urbain mais nos solutions possèdent d'autres caractéristiques. Ainsi, notre îlot n'est pas limité à un rectangle quadrillé. Tout type de pavage avec des polygones peut modéliser un îlot. Cela permet de tester des bâtiments avec des formes plus variées et des autres propriétés sur leur périmètre et leur aire. Notons également que l'utilisateur peut spécifier les dimensions de l'îlot ainsi que la taille des cellules qui le composent. Enfin, une dernière fonctionnalité est de pouvoir verrouiller des cellules avant de lancer le programme. Ceci

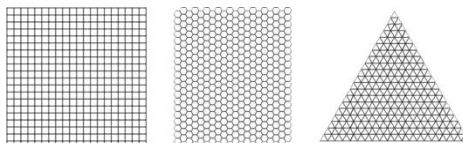


Figure 1. Quelques exemples de pavages

nous permet de modéliser la présence de bâtiments historiques dans notre îlot ou d'espaces verts protégés, *i.e.* des zones que le programme ne peut en aucun cas modifier.

Le voisinage d'une solution donnée est l'ensemble des solutions que nous pouvons obtenir en changeant la couleur d'une cellule donnée. Cela peut avoir comme effet de créer un nouveau bâtiment, agrandir un bâtiment existant, supprimer un bâtiment si nous changeons sa couleur en blanc ou simplement changer la couleur d'un bâtiment. Changer simplement la couleur d'un bâtiment peut être intéressant car deux bâtiments suffisamment proches avec la même couleur peuvent fusionner pour en former un plus grand. Nous négligeons dans un premier temps la notion de compacité de l'îlot urbain pour nous concentrer sur la notion de *porosité* (rapport entre l'aire non-bâtie et l'aire totale de l'îlot). Notre objectif est de minimiser la porosité.

Comme expliqué précédemment, la recherche locale peut prendre en compte des contraintes. Dans notre cas, les contraintes guident le programme afin d'obtenir des solutions plus réalistes et intéressantes. En effet, avec la porosité minimale comme unique objectif, le programme peut retourner comme solution un îlot urbain rempli uniquement avec un seul gros bâtiment ou avec des centaines de petits bâtiments sans espaces entre eux. Ce genre de solutions ne respectent pas nos contraintes portant sur le confort et le respect de la vie privée, elles doivent donc être évitées.

La première contrainte obligatoire que nous avons implémentée garantit une distance minimale entre les façades des différents bâtiments. Les autres contraintes sont des contraintes pénalisantes. Elles pénalisent les solutions si le nombre de bâtiments, le périmètre total, le périmètre de chaque bâtiment ou l'aire de chaque bâtiment n'appartiennent pas à un certain intervalle.

2.4 Analyse de certains résultats

L'implémentation de la méthodologie de l'outil dynamique nécessite de nombreux tests. Les observations et les analyse de ces résultats nous guident dans nos choix de considérations et nous conduisent vers de nombreuses possibilités.

La Figure 2 illustre un résultat obtenu par le programme sur un îlot de 102 m par 102 m pavé par des carrés de 2 m de côtés. L'objectif est de minimiser la porosité. En plus, nous prenons en compte les contraintes suivantes : avoir entre 6 et 10 bâtiments, les bâtiments doivent avoir un périmètre compris entre 75 m et 500 m et il doit y avoir minimum 4 m (ou deux carrés vides) entre les façades de deux bâtiments. Chaque bloc de cellules avec le même motif représente un bâtiment.

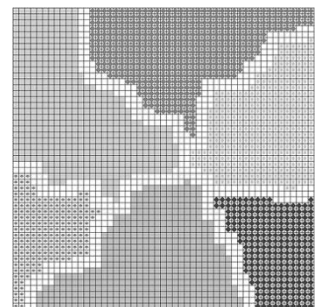


Figure 2.

La contrainte sur le nombre de bâtiments permet d'éviter un îlot urbain avec plein de petits bâtiments. La contrainte sur le périmètre permet d'éviter les résultats avec un gros bâtiment et quelques petits. Comme l'objectif est de minimiser la porosité, l'outil veut colorer le plus de cellules possibles dans le pavage. Les résultats tendent donc vers un îlot urbain rempli. La contrainte portant sur la distance entre les façades empêche cela. Comme observé à la Figure 2, cette contrainte crée des espaces vides qui peuvent être interprétés comme des chemins, des voies de circulation dans notre îlot.

Seulement considérer des bâtiments peut donc mener à la création de voies de circulation dans notre îlot. Cependant, nous n'avons aucun contrôle sur ces voies et leurs propriétés. De plus, nous voulons un certain pourcentage d'espaces verts dans notre solution (De Smet, 2018) et il n'y a actuellement plus de place pour eux. À ce stade, nous avons besoin d'introduire d'autres composantes de l'îlot, en particulier, les espaces verts.

Afin de distinguer les espaces verts des bâtiments, ceux-ci sont représentés par le motif + dans nos Figures. Comme mentionné en introduction, nous avons des contraintes sur les espaces verts :

- viser un certain pourcentage d'espaces verts sur la surface de l'îlot urbain,
- garantir qu'au maximum deux espaces verts forment 50% de la totalité des espaces verts.

Pour plus de flexibilité, nous avons implémenté la seconde contrainte de façon générale. L'utilisateur peut spécifier le pourcentage et combien d'espace doivent former ce pourcentage.

Nous avons actuellement intégré les espaces verts d'une façon assez naïve. Durant la recherche locale, le programme a simplement une nouvelle couleur : le motif +. Avec cette approche, nous obtenons des résultats comme celui de la Figure 3. Cette solution est retournée avec des contraintes similaires à la Figure 2. Nous avons juste demandé un nombre de bâtiments entre 6 et 9 au lieu de 6 et 10. Nous imposons également 20% d'espaces verts dont 50% sont formés par au plus deux espaces verts.

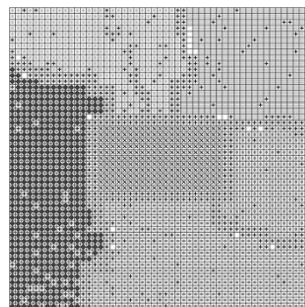


Figure 3.

Nous observons à la Figure 3 que l'outil a juste rempli les voies vides formées par la distance entre les façades avec des espaces verts. Ensuite le programme a créé plusieurs espaces verts éparpillés afin d'atteindre les 20% ciblés. Cette solution est problématique car, entre autres, elle ne nous laisse plus la possibilité de créer des voies de circulations. Ce genre de solution apparaît parce que minimiser la porosité et viser une certaine proportion d'espaces verts est source de conflits. D'un côté, nous voulons la plus grande aire bâtie possible ; mais de l'autre, nous voulons moins de bâtiments pour créer des espaces verts. Pour éviter les conflits, l'outil place des espaces verts dans les cellules qui ne peuvent être bâties en priorité. Ensuite, il essaie de placer des espaces verts où il peut afin d'atteindre les 20%.

De plus, à partir des premiers essais, les simulations (voir Figure 2) montrent des résultats très découpés à l'intérieur du bloc lui-même tandis que les limites périphériques des bâtiments créés suivent toujours les limites périphériques de l'îlot (carrées ici). Permettre au programme de retourner uniquement ce genre de solutions risque de standardiser les formes à l'extérieur de l'îlot. À ce stade de notre étude, nous avons décidé de négliger cette problématique. Néanmoins, observer les simulations à différentes échelles (à l'échelle de l'îlot mais aussi du quartier et pourquoi pas à l'échelle de la ville) permettrait de résoudre ce problème. L'outil pourrait permettre de travailler sur différentes échelles successives. A l'échelle du quartier, l'outil déterminerait une coupe d'une zone (quartier) en îlots urbains aux formes variées. A l'échelle de l'îlot, il en étudierait son contenu.

3. Un modèle multi-objectifs basé sur la théorie des jeux

Les premiers résultats obtenus avec deux composantes de notre îlot urbain nous amènent à discuter des limitations de notre premier modèle. Comme expliqué auparavant, le programme essaie de minimiser son objectif (la porosité de l'îlot) et d'être le moins pénalisé possible. Pour réaliser cela, le programme hiérarchise l'objectif et les contraintes. Plus la pénalité est grande, plus elle est "importante". Avec cet ordre, une contrainte peut être totalement négligée par le programme parce qu'elle est plus faible qu'une autre. Cela rend l'introduction d'une nouvelle contrainte difficile et ennuyante si nous voulons éviter la hiérarchisation des contraintes. De plus, il n'est pas aisé d'implémenter différents comportements pour chaque composante. Or, il peut être pertinent que le programme modifie chaque composante d'une façon différente. C'est pourquoi nous nous intéressons maintenant à des modèles multi-objectifs, en particulier aux modèles issus de la *théorie des jeux* (Lã, Chew, & Boon-Hee, 2016).

La théorie des jeux est un domaine des mathématiques permettant, entre autres, de modéliser les interactions entre des systèmes et plusieurs agents via des *jeux*. Nous allons nous pencher en particulier sur les jeux *sous forme stratégique*. Pour définir un tel jeu, nous avons besoin d'un ensemble de n joueurs numérotés de 1 à n (où n est un entier naturel non nul). Chaque joueur possède un ensemble de stratégies, d'actions qu'il peut choisir. Quand nous formons une séquence avec une stratégie de chaque joueur (ordonnée en suivant l'ordre des numéros des joueurs), nous obtenons un *profil de stratégies*. Chaque joueur possède une *fonction de gains*, fonction dépendant des différents profils de stratégies possibles. L'objectif de chaque joueur est de maximiser ou de minimiser sa fonction de gain. En plus de cela, nous supposons que chaque joueur est *égoïste* et *rationnel*. Ainsi un joueur va toujours chercher à optimiser son profit, qu'importe les conséquences.

Le Tableau 1 est un petit exemple de jeu. Nous avons ici un jeu à deux joueurs, J_1 et J_2 . Les deux joueurs ont trois stratégies A , B et C . Les stratégies verticales sont celles de J_1 et les horizontales celles de J_2 . Un tel jeu se joue à la façon du "pierre-feuille-ciseaux" *i.e.*, les joueurs choisissent en même temps leur stratégie et reçoivent leurs gains. Ainsi, si le profil choisi par les deux joueurs est (A, B) , J_1 reçoit un gain de -5 et J_2 un gain de 9 .

Tableau 1. Un exemple de jeu

	A	B	C
A	(3, 3)	(-5, 9)	(-3, 8)
B	(9, -5)	(2, 2)	(2, 6)
C	(6, 3)	(7, 7)	(5, 5)

Parmi les profils de stratégies, nous nous intéressons particulièrement à certains, les *Équilibres de Nash* (EN). Un EN est un profil de stratégies tel que pour tout joueur son action est une meilleure réponse (un meilleur choix) face aux choix des autres joueurs *i.e.*, son action le meilleur gain en tenant compte des stratégies des autres joueurs. Dans l'exemple du Tableau 1, le profil (C, B) est un EN. Si un seul joueur changeait de stratégie, il gagnerait moins.

Avec les EN viennent deux questions. Étant donné un jeu, existe-t-il un EN ? Si oui, comment le trouver ? Afin d'y répondre, nous nous sommes penchés sur des jeux particuliers, les *Potential Games*. Nous ne donnons pas tous les détails ici mais ces jeux nous fournissent des résultats qui garantissent l'existence d'EN et l'arrêt de la méthode pour les trouver que

nous expliquerons ci-après. Le lecteur intéressé peut trouver tous les détails dans l'ouvrage (Lâ, Chew, & Boon-Hee, 2016).

Pour trouver un EN, il existe une technique très similaire à la recherche locale dans son idée. En démarrant d'un profil de stratégie quelconque, nous permettons à chaque joueur de successivement changer sa stratégie afin d'améliorer son gain. Nous nous arrêtons quand plus aucun joueur ne peut améliorer son gain, *i.e.* quand nous trouvons un EN.

Maintenant que nous avons défini notre nouvelle méthode, expliquons comment nous essayons de l'appliquer à notre problème urbanistique. À chaque joueur est attribué un critère de compacité qu'il va chercher à optimiser ou à satisfaire. Dans notre jeu, les profils de stratégies sont des configurations de l'îlot. Maintenant, nos joueurs sont des critères de compacité qui vont chercher à modifier des configurations d'îlots. Comme nous définissons nous-mêmes comment les joueurs modifient les configurations, en étant attentifs et pointilleux nous pouvons satisfaire les conditions qui nous garantissent de trouver une configuration correspondant à un EN.

La Figure 4 illustre un résultat obtenu à la suite d'une partie entre quatre joueurs. Le premier joueur vise un certain nombre de bâtiments dans l'îlot. Le second est l'équivalent du premier mais pour les espaces verts. Le troisième veut que le périmètre de chaque bâtiment appartienne à une certaine fourchette. Enfin, le quatrième vise un certain pourcentage d'espaces verts. Les stratégies des deux premiers joueurs consistent à simplement changer la couleur d'une cellule. Les deux autres ont la possibilité de colorier toutes les cellules adjacentes aux bords des composants de l'îlot qui les concernent. Notons que nous n'avons pas ici de joueurs dont l'objectif est de minimiser la porosité. Cependant, c'est un objectif sous-jacent du troisième joueur.

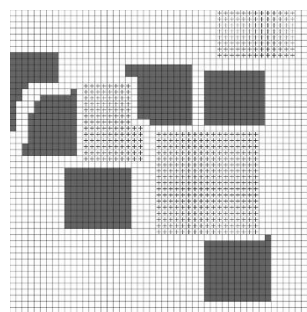


Figure 4.

Les premiers résultats obtenus avec cette approche tendent à résoudre certains problèmes rencontrés avec les solutions retournées par les algorithmes de recherche locale. En effet, nous observons à la Figure 4 que les espaces verts ne sont plus totalement éparpillés et ont maintenant une forme compacte. De plus, les bâtiments ne suivent maintenant plus la périphérie de l'îlot. Il reste malgré tout un travail à fournir afin de trouver les paramètres qui amènent aux meilleures configurations et afin d'introduire plus de composantes de l'îlot.

4. Conclusion

Comme expliqué auparavant, notre outil d'aide à la conception d'îlots compacts a pour but de minimiser la porosité de l'îlot tout en respectant un ensemble de contraintes. Les premiers résultats obtenus à l'aide des algorithmes de recherche locale sont prometteurs mais peuvent être améliorés, notamment la gestion des espaces verts. C'est pourquoi nous avons adopté une nouvelle approche basée sur la théorie des jeux. Cette approche permet à chaque contrainte et à chaque critère de modifier la configuration de l'îlot à son avantage. Nous cherchons ainsi à éviter une hiérarchisation des contraintes. Cela nous permet d'implémenter différents comportements à chacune des composantes de l'îlot. De plus, les solutions obtenues avec cette approche tendent à résoudre les problèmes rencontrés avec la recherche locale.

Même si, dans un premier temps, nous avons négligé la notion de compacité, nous restons conscients que c'est notre principal objectif. Notre modèle est fréquemment enrichi de nouvelles contraintes visant à atteindre la compacité de l'îlot et des solutions plus réalistes.

Pour conclure, rappelons que les modèles que nous avons présentés dans ce papier sont très généraux. Nous avons montré ici une application spécifique avec une modélisation spécifique. Ainsi, la méthodologie que nous utilisons peut être appliquée à divers autres problèmes.

Bibliographie

- Arantes, L., Marry, S., Baverel, O., & Quenard, D. (2016). Efficacité énergétique et formes urbaines : élaboration d'un outil d'optimisation morpho-énergétique. *Cybergeog : European Journal of Geography [En ligne]*, 777, p. 29.
- Bertaud, A., & Malpezzi, S. (2003). *The Spatial Distribution of Population in 48 World Cities : Implications for Economies in Transition*. Consulté le juillet 3, 2017, sur www.alain-bertaud.com: http://alain-bertaud.com/AB_Files/Spatia_%20Distribution_of_Pop_%2050_%20Cities.pdf
- Bonin, O., & Tomasoni, L. (2013). *Rendre la ville plus compacte : réflexion autour d'un scénario alternatif à l'augmentation des densités*.
- Clément, P., & Guth, S. (1995). De la densité qui tue à la densité qui paye. La densité urbaine comme règle et médiateur entre politique et projet. *Densités et espacements - Les annales de la recherche urbaine*, 67(1), pp. 72-83.
- Dantzig, G.-B., & Saaty, T.-L. (1973). *Compact City : Plan for a Liveable Urban Environment*. San Francisco: W.-H. Freeman.
- De Beukelaer, H., Davenport, G. F., De Meyer, G., & Fack, V. (2017). JAMES: An object-oriented Java framework for discrete optimization using local search metaheuristics. *Software: Practice and Experience*, 47(6), 921-938.
- De Smet, I. (2018). *Elaboration et expérimentation d'un outil d'évaluation et d'aide à la conception compacts à dominante d'habitat suivant une densité de population cible*. UMONS, Mons.
- De Smet, I., & Laplume, D. (2019). Design of compact residential blocks for sustainable urban regeneration: determination of consistent qualitative criteria. *Journal of Urban Regeneration & Renewal*, 248-257.
- El-Ghazali, T. (2009). *Metaheuristics: from design to implementation*. John Wiley & Sons.
- Frankhauser, P., Tannier, C., Vuidel, G., & Houot, H. (2008). Une approche multi-échelle de l'accessibilité pour maîtriser l'étalement urbain. *International Conference on Mobility and Transport* (p. 18). München: Technische Universität München - Institute for Transportation.
- Grenier, A. (2007). Ville et énergie. Spécificité et complexité de la question en France. Dans A. Grenier, & M.-F. Mattei, *La ville dans la transition énergétique - Les annales de la recherche urbaine* (Vol. 103, pp. 131-138).
- Halleux, J.-M. (2012, 1-2). Vers la ville compacte qualitative? Gestion de la périurbanisation et actions publiques. *Belgeo*.
- Jenks, M., Burton, E., & Williams, K. (1996). *The Compact City : A Sustainable Urban Form?* Spon Press.
- Katz, P. (1993). *The New Urbanism : Toward an Architecture of Community*. New York: McGraw-Hill Professional.
- Kirwan, R. (1992). Urban form, energy and transport : A note on the Newman-Kenworthy thesis. *Urban Policy and Research*, 10(1), pp. 6-22.
- Lã, Q. D., Chew, Y. H., & Boon-Hee, S. (2016). *Potential Game Theory*. Springer.

- Maignant, G. (2005). Compacité et forme urbaine, une analyse environnementale dans la perspective d'un développement durable. *Développement urbain durable, gestion des ressources et gouvernance*, (p. 17). Lausanne.
- Marique, A. (2013). *Méthodologie d'Evaluation Energétique des Quartiers Périurbains. Perspectives pour le Renouveau Périurbain Wallon*. . Liège: ULiège, FSA, ArGENCO-LEMA.
- Newman, O. (1975). An ecological model for city structure and development. *Ekistics*(239), pp. 258-264.
- Newman, P., & Hogan, T. (1981). A Review of Urban Density Models : Toward a Resolution of the Conflict Between Populace and Planner. *Human Ecology*, 9(3), pp. 269-303.
- Pouyane, G. (2004). "Des avantages comparatifs de la ville compacte à l'interaction forme urbaine-mobilité. Méthodologie et premiers résultats.". *"Les cahiers scientifiques du transport"*(45/2004), p. 53.