À LA RECHERCHE D'UN DESIGN URBAIN ORIGINAL

A PREPRINT

Valentin. Meo
Department of Geomatics
Laval university
Quebec, QC
valentin.meo.1@ulaval.ca

Théo. Cavaillès
Département d'informatique et de génie logiciel
Laval university
Quebec, QC
theo.cavailles.1@ulaval.ca

December 20, 2022

1 Introduction

Le design des villes a un impact certain sur le bonheur de ses habitants. De plus, un mauvais design conduit à des coûts évitables pour ses habitants et ses entreprises, par exemple dus aux transports ou à l'exposition à certains risques naturels ou sanitaires. Bien que le sujet soit connu, l'absence de moyens ou de personnes formées est souvent un problème.

De nombreuses solutions existent pour aider à la prise de décision en urbanisme, par exemple ArcGIS Urban [1]. Mais ces solutions n'ont pas vocation originale à être des moteurs de recommandation. Elles ne sont pas totalement automatiques et surtout tiennent peu compte des états postérieurs pour créer des recommandations. Or les méthodes d'apprentissage par renforcement tiennent compte des états postérieurs pour prendre la décision à un instant donné. De plus, elles peuvent amener des designs urbains plus créatifs. Enfin, cela pourrait revenir moins cher et facile d'utilisation.

Ce rapport va explorer et analyser de manière rigoureuse l'utilisation de méthodes d'apprentissage par renforcement pour produire un moteur de recommandation de zonning créatif, génératif et efficace avec un exemple concret.

2 Related work

L'urbanisme est défini comme un processus technique et politique axé sur le dévelopement et la conception de l'utilisation des espaces et de l'environnement bâti. Ainsi, dans le grand ensemble de l'urbanisation, les domaines techniques comme la construction et l'arpentage sont les plus matures. La construction est de loin l'industrie la plus explorée : cela va d'algorithmes de planification de chantier [2], à l'automatisation des machines [3], à la surveillance vidéo [4]. Cependant, la recherche en planification urbaine aidée par l'apprentissage par renforcement est un domaine encore peu exploré. Des recherches ont cependant été réalisées.

Deux travaux similaires [5, 6] explorent comment un agent peut organiser les bâtiments d'un bloc dans un environnement 3D. L'agent peut choisir la position du bâtiment, sa hauteur, sa longueur et sa largeur et a pour but principal de maximiser directement ou indirectement la luminosité. L'agent utilisé est de type DDPG pour un papier et DQN pour l'autre. Le résultat du travail conduit à des organisations de quartiers créatives et réalistes. Un autre type de travail, en lien avec l'organisation des villes, a été réalisé dans le but de résoudre le jeu vidéo de construction SimCity. En effet, les jeux ont un environnement complexe déjà programmé avec des rewards faciles à mettre en place. Ce sont donc des problèmes facilement transposables à l'apprentissage par renforcement. Bien que le travail [7] ne conduise pas à des resultats satifaisants, les resultats montrent des signes positifs, qui demontrent la capacité des approches par renforcement pour planifier le zonage des villes. Un autre travail [8] propose une méthode pour modéliser le système routier via l'apprentissage supervisé. Ce travail exploite une architecture multi-agent, avec une approche de type TD. Bien que ce travail soit peu pratique, il montre que l'apprentissage par renforcement peut arriver à des sytèmes de modélisation cohérents pour le réseau routier en résolvant certains problèmes.

Cependant, aucune recherche sur l'utilisation du *renforcement learning* appliqué à la conception de zonage, de l'aménagement, de planification du développement d'une ville ou au design en général n'a été trouvé.

3 Méthodes

3.1 Environnement

Dans le but de démontrer les possibilités du *renforcement learning* pour produire un moteur de recommandation, un environnement simple a été implémenté. L'environnement représente plus particulièrement le zonage d'une ville en damier, chaque cellule étant une unité de zonage Figure 1. Sur cette figure les couleurs représentent des zonages différents.

Les types de zonage implementés sont les zones résidentielles, les zones de travail parfois nommées "bureaux", les zones commerciales et les zones de divertissement parfois nommé "parcs" Figure 1.

Chaque épisode commence par une zone aléatoire de taille 3x3. À chaque pas de temps, l'agent affecte un usage à une cellule du damier parmi ceux représentés Figure 1. Les zonages sont affectés à la bordure de la ville jusqu'à compléter le plus petit carré englobant. Un épisode s'arrête après 300 itérations.

L'espace d'observation est un carré de 6x6 cellules avec la cellule à définir au centre. La reward est évaluée dans un carré de même type mais de taille 5x5. Dans la mesure où les circonvolutions se réalisent de la même façon, cela permet à l'agent de conserver les propriétés markoviennes.

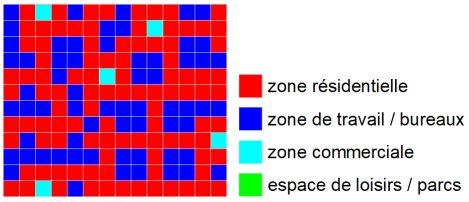

Figure 1: À gauche un exemple d'environnement, à droite les zones et l'espace des actions et leur couleur respective

La fonction de reward est définie par les formules (1), (2), (3) et (4). Le but est de produire un moteur de recommandation avec des facteurs de pondération qui sont ajustables au besoin unique de chaque ville. Le choix des facteurs actuels à été motivé d'abord par notre intuition, puis ajustées et validées avec quelques tests. Pour mesurer la possibilité de l'agent à générer des villes créatives, la reward doit être équilibrée, sans surpondération trop importante sur un type de zonage mais en s'approchant d'une diversité réaliste ; c'est-à-dire 50% habitations, 25% bureaux et 25% commerces et parcs. Les fonctions ont été choisies avec une approche intuitive sur la valeur des différents zonages. Par exemple, une maison a plus de valeur proche d'un parc.

La reward des maisons :

$$x\frac{1}{\sqrt{dist\ nearest\ office}^2} + y\frac{nb\ adjacent\ houses}{8} + z\frac{1}{\sqrt{mean\ dist\ shop}^2} + k\frac{1}{\sqrt{dist\ nearest\ park}^2} \tag{1}$$

$$x = 5,\ y = 1,\ z = 7\ et\ k = 7$$

La reward des bureaux :

$$x\frac{1}{\sqrt{mean\ dist\ house}^2} + y\frac{1}{\sqrt{dist\ nearest\ office}^2} - z \times r_1 \tag{2}$$

$$x = 5,\ y = 1\ et\ z = 1$$

$$r_1 = \begin{cases} -1 & if\ office\ in\ the\ buffer\ >\ 30\%\ of\ non-empty\ cases \\ 1 & else \end{cases}$$

La reward des parcs :

$$x\frac{nb\ houses}{5} \tag{3}$$

$$x = 1$$

La reward des commerces :

$$x\frac{nb\ adjacent\ houses}{1} \tag{4}$$

$$x = 1$$

3.2 Algorithme

La littérature a montré que DQN obtient des résultats génératifs [6]. En effet, les méthodes de deep RL sont parfaites pour obtenir des résultats génératifs et créatifs. De plus, le nombre d'etats distincts étant relativement grand, les méthodes plus traditionnelles qui n'utilisent pas d'approximation de fonctions ne sont pas réalisables dans un temps d'entraînement raisonnable. Nous allons donc ici utiliser des méthodes de type Deep RL.

Le but de la recherche étant de montrer un potentiel, les tests se sont concentrés sur des modèles communs et connus pour leur robustesse empirique. Ainsi, nous avons choisi d'utiliser la bibliothèque la plus courante : stable baseline [9] et ses modèles vitrines. À savoir un simple DQN, modèl incontournable qui a montré des résultats sur des problématiques similaires dans la littérature, PPO pour tester une approche *policy based*, et A2C qui pourrait produire des résultats plus génératifs grâce à la nature moins convergente des méthodes *adversarial*de type *actor critic*. Cela a été montré avec les approches de type Gan en apprentissage supervisé.

Afin de comparer ces résultats, nous avons choisi d'utiliser 2 baselines. Tout d'abord, une politique aléatoire puis une politique naïve myope optimale. C'est-à-dire que cette dernière méthode va calculer toutes les possibilité et selectionner celle qui maximise la récompense immédiate.

4 Expérience et résultats

4.1 Expérience

Les algorithmes metionnés dans la Section 3.2 ont ainsi été testés sur l'environnement discuté dans la Section 3.1. L'entraînement a été réalisé sur 30 épisodes et les résultats de ces expériences sont présentés dans la section suivante. D'autres expériences ont été réalisées en jouant notamment sur quelques subtilités, mais aussi avec ce qui avait été remarqué dans la correction de la vidéo. Malheureusement, aucun de ces changements n'a amené à une amélioration significative de la performance. Le code et les détails de l'implementation sont disponibles en annexe de ce document et sur github [1].

4.2 Résultats

4.2.1 Résultats quantitatifs

Les résultats de l'expérience décrite dans la section précédente sont montrés Figure 2 où l'ordonnée représente la reward cumulative sur tout l'épisode pendant l'entraînement. L'évolution de la diversité de la prédiction pendant l'entraînement est montré Figure 3.

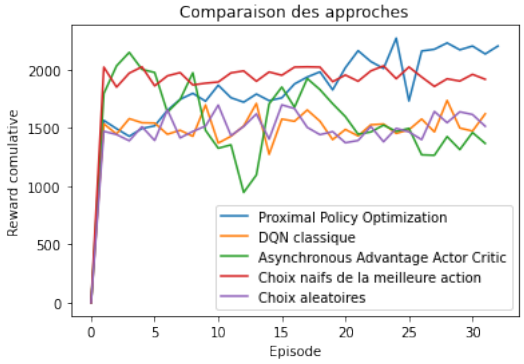

Figure 2: Reward cumulative par épisode pendant l'entraînement

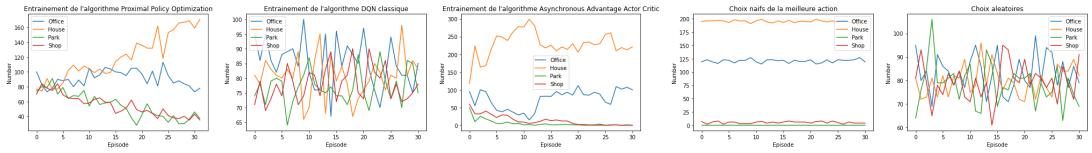

Figure 3: Diversité de zonage observée par épisode pendant l'entraînement

4.2.2 Résultats qualitatifs

Cette section présente les résultats qualitatifs obtenus avec l'expérience décrite précédemment. Les figures suivantes représentent les villes produites par les modèles à la fin de leur entraînement. La Figure 4 représente les résultats avec les méthodes de baseline et la Figure 5 représente les résultats des modèles de Deep RL.

5 Discussion et Conclusion

5.1 Discussion

Les résultats de la baseline sont intéressants. En effet, l'analyse de la diversité Figure 3 et l'analyse qualitative Figure 4 montrent que le modèle optimal n'arrive pas à prédire de parc. En effet, construire un parc sur la bordure d'une ville apporte peu de reward immédiate, mais il apporte de la valeur à toutes les habitations qui seront prochainement

[1] https://github.com/valkenzz/GenerativeCityPlaningWithDeepRL

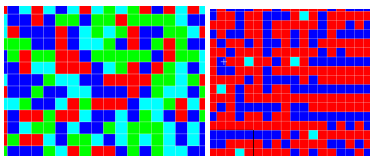

Figure 4: Résultats qualitatif des baselines : à gauche le modèle aléatoire et à droite le modèle myope optimal

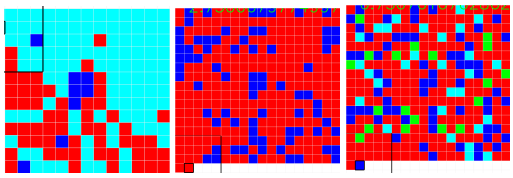

Figure 5: Résultats qualitatifs des modèles après entraînement: à gauche DQN, au centre A2C et à droite PPO

construites autour. On observe donc ici un comportement typiquement myope, qui préfère une reward immédiate faible à une reward future importante.

Les performances de DQN sont assez faibles. En effet, la Figure 2 montre que DQN a le même ordre de grandeur de reward cumulée que la baseline aléatoire. La Figure 3 montre que la diversité est très similaire au cas aléatoire. Enfin, l'analyse qualitative montre que DQN est très instable, car la Figure 5 montre une ville majoritairement commerciale alors que dans les analyses de diversité, pendant l'entraînement, la ville est quasiment équi-ponderée pour chaque type de zone. On peut en conclure que le réseau DQN change de prédiction à chaque itération de l'entraînement. Pour résoudre ce problème, des Add-on, qui ont pour but d'améliorer la stabilité, pourront être rajoutées. Un réseau de neurones de type convolution pourrait aussi aider étant donné la nature spatiale du problème.

Les performances A2C sont également faibles. En effet, la Figure 2 montre que les rewards obtenues sont les mêmes qu'avec une politique aléatoire. On peut toutefois noter qu'au début de l'entraînement, la reward sur deux épisodes a outperformé la baseline optimale. De plus, la Figure 3 montre que la diversité est très différente du cas aléatoire, ce qui est bon signe. La Figure 5 montre cependant que la ville est très simple, on peut donc supposer que le modèle est bloqué dans un minimum local et qu'il n'a pas bien appris la valeur des parcs et des centres commerciaux. Cela vient probablement de la la nature adversarialisée d'A2C qui dépend d'un réseau critique de type DQN pour obtenir des résultats convenables. Or, les résultats de DQN montrent que la fonction de valeur n'est pas approximée correctement.

Enfin PPO est l'agorithme qui a eu les meilleurs résultats. En effet, la Figure 2 montre que PPO s'améliore lentement et de façon stable et, vers la fin de l'entraînement, outperform la baseline optimale. De plus, la Figure 3 montre que, malgré une diversité de départ aléatoire, la diversité finale est proche des résultats attendus : c'est-à-dire proche d'une ville équilibrée avec majoritairement des habitations, un peu moins de bureaux et quelques centres commerciaux et parcs. Enfin, l'analyse qualitative nous a montré que le modèle produit un design créatif, génératif et original. Certaines incohérences sont à noter et dans la mesure où l'évolution de la reward cumulative n'avait pas l'air de plafonner, on peut supposer qu'un entraînement plus long donnerait de meilleurs résultats.

Finalement, il convient de noter que la méthode d'entraînement n'est pas optimale. Tout d'abords, elle ne prend pas en compte tout l'espace pour alléger les calculs. De plus, plus le carré et grand, plus de temps d'une circonvolution est long, donc plus l'impact de la reward postérieure va être petit. La possibilité d'apprendre des états postérieurs est donc limitée.

5.2 Conclusion

Ainsi, les résultats ont montré qu'un algorithme de RL surperforme la méthode myope optimale, aussi bien d'un point de vue quantitatif que qualitatif. Grâce à l'analyse qualitative, on a même montré que PPO fait preuve de créativité et produit une ville générative. On a ainsi montré le potentiel de ces méthodes de RL dans la recherche de designs urbains originaux, plus particulièrement pour le zonage. Cela ouvre plusieurs perspectives, notamment entraîner plus longtemps PPO, ce qui na pas pu être réalisé dans cette étude par manque de ressources calculatoires. La stabilité de DQN et, indirectement, de A2C peut également être améliorée avec l'exploration des nombreux Add-on. L'utilisation de réseaux de type CNN ou d'approximateurs de fonctions de *deep decision trees* pourrait être appropriée. Enfin, une amélioration de l'environnement serait intéressante à explorer, notament avec l'ajout d'un système routier [8] et une modification de l'entraînement. Un moteur de recommandation paramétrable est donc quelque chose de réalliste qu'il ne reste plus qu'à implémenter ;).

References

[1] Esri, ArcGIS Urban, URL [consulted 12-20-2022]: *https://www.esri.com/en-us/arcgis/products/arcgis-urban/overview*.

[2] Taehoon Kim, Yong-Woo Kim, Dongmin Lee, Minju Kim, Reinforcement learning approach to scheduling of precast concrete production, In *Journal of Cleaner Production, Volume 336, 2022, 130419, ISSN 0959-6526,* DOI : https://doi.org/10.1016/j.jclepro.2022.130419.

[3] Xu, Xinghui & García de Soto, Borja, Reinforcement learning with construction robots: A review of research areas, challenges and opportunities, *2022,* DOI : https://doi.org/10.22260/ISARC2022/0052.

[4] Gabriel Ramos & Kevin Laurent, Computer-Aided Labelling for Inspection (CALI) powered by Reinforcement Learning, URL [consulted 12-20-2022] : *https://www.youtube.com/watch?v=Re_g_QE20WA&list=PLl9u2iHFOwRE9YBqHRayaGNod4fSJOPCT&ab_channel=AudreyDurand*.

[5] Han, Z., Yan, W., Liu, G., A Performance-Based Urban Block Generative Design Using Deep Reinforcement Learning and Computer Vision, In *Yuan, P.F., Yao, J., Yan, C., Wang, X., Leach, N. (eds) Proceedings of the 2020 Digital FUTURES. CDRF 2020. Springer, Singapore,* DOI : https://doi.org/10.1007/978-981-33-4400-6_13.

[6] Chenyu, H., Gengjia, Z., Miggang, Y., Jiawei, Y., Energy-Driven Intelligent Genrative Urban Design : Based on deep reinforcement learning with a nested Deep Q-R Network, In *POST-CARBON, CAADRIA 2022,*URL [consulted 12-20-2022] : https://caadria2022.org/wp-content/uploads/2022/04/239-1.pdf.

[7] smearle, gym-city, On github, URL [consulted 12-20-2022] : https://github.com/smearle/gym-city.

[8] Mortaza Zolfpour-Arokhlo, Ali Selamat, Siti Zaiton Mohd Hashim, Hossein Afkhami, Modeling of route planning system based on Q value-based dynamic programming with multi-agent reinforcement learning algorithms, In *Engineering Applications of Artificial Intelligence, Volume 29, 2014, Pages 163-177, ISSN 0952-1976,* DOI : https://doi.org/10.1016/j.engappai.2014.01.001.

[9] Welcome to Stable Baselines docs! - RL Baselines Made Easy © *Copyright 2018-2021, Stable Baselines Revision 550db0d6.* URL [consulted 12-20-2022] : https://stable-baselines.readthedocs.io/en/master/.