Couplage de simulations multi-agents pour la conception de politiques urbaines

S. Pageaud[a,b] simon.pageaud@liris.cnrs.fr
V. Deslandres[a] veronique.deslandres@liris.cnrs.fr
S. Hassas[a] salima.hassas@liris.cnrs.fr
V. Lehoux[b] vassilissa.lehoux@naverlabs.com

[a]Laboratoire LIRIS, Université Claude Bernard Lyon 1, Lyon, France

[b]NAVER LABS Europe, Meylan, France

Résumé

Dans un futur proche, la disponibilité croissante des données imposera aux décideurs politiques de modifier régulièrement les politiques urbaines afin d'intégrer l'évolution des comportements et les retours utilisateurs. Dans ce papier, nous proposons une architecture multi-agent générique permettant de concevoir et de modéliser des politiques urbaines afin d'en éprouver la pertinence en la déployant sur un environnement spécifique. Ces environnements sont conçus en exploitant des données provenant de n'importe quelle ville disposant de données ouvertes et communautaires (Open Street Map). Deux modèles multi-agents sont couplés dans une boucle dynamique micro-macro et peuvent être modifiés à la fois par des techniques d'apprentissage par renforcement ainsi que par l'intégration du retour des décideurs politiques. Nous proposons une formalisation permettant de représenter les politiques urbaines pour initier une co-construction entre le décideur politique et notre système.

Une expérimentation sur la régulation de la tarification d'emplacements de stationnement en zone urbaine permet de justifier l'usage de notre architecture pour concevoir des politiques urbaines pertinentes.

Mots-clés : *Modélisation multi-agents ; Simulation multi-agents ; Conception de politiques urbaines ; Apprentissage par renforcement.*

Abstract

In the near future, the increasing availability of data will require policy makers to regularly change urban policies to incorporate changing behavior and user feedbacks. In this paper, we propose a generic agent-based architecture for designing and modeling urban policies in order to test their relevance by deploying them on a specific environment. Environments are designed using data from the city provided by any available open data toolkit (Open Street Map here). Two agent-based models are coupled in a micro-macro dynamic loop and they can be adapted either by the system using reinforcement learning or by the stakeholders using simulation results. We propose a formalism to represent urban policies to allow a co-designed iterative process between the policymaker and our system.

An experimentation on the regulation of parking prices in downtown area justifies the use of our architecture to design relevant urban policies.

Keywords: *Multi-agents simulation ; Social simulation ; Smart city ; Urban Policymaking ; Reinforcement Learning.*

1 Introduction

1.1 Contexte

Ce papier propose une approche générique pour la conception de politiques urbaines, dans un horizon proche où l'infrastructure de la ville sera équipée de moyens étendus d'action (véhicules connectés, actionneurs dynamiques sur les voies) et fournissant un grand nombre de données sur le trafic et l'état du réseau routier. Dans la smart city de demain, les parties prenantes, que ce soit les décideurs politiques ou les usagers, auront besoin de concevoir des politiques urbaines dans des environnements en évolution continue [4]. Afin de faciliter l'acceptation des politiques à la fois par les utilisateurs et par les décideurs, les politiques doivent évoluer avec les changements de l'environnement et intégrer le retour des décideurs et des utilisateurs pour en améliorer la qualité [8]. Les politiques urbaines doivent également être robustes ou s'adapter à la population. La pertinence de nombreuses politiques dépend de la population : la même politique peut mener à des résultats opposés dans

deux villes différentes ou même deux quartiers d'une même ville [6].

L'élaboration de politiques urbaines pertinentes nécessite une bonne compréhension du processus de création global : la différence entre l'idée et l'implémentation d'une politique, l'impact sur les personnes concernées, le budget des décideurs. Cependant, si une politique n'est pas pertinente ou refusée par les utilisateurs, alors le temps et l'argent investis dans la collecte de données, l'étude de marché, l'implémentation et l'étude de la pertinence sont perdus [7]. Ainsi la simulation réaliste –basée sur une infrastructure réelle, avec des modèles d'agents représentatifs de la population réelle– offre une perspective intéressante pour le décideur.

Le processus de création de politique est généralement découpé en deux parties : la conception et l'analyse. L'utilisation de systèmes complexes et adaptatifs est un verrou concernant la création de politiques, ce qui explique pourquoi les approches classiques et prédictives peinent à améliorer leur efficacité [2, 4]. C'est pourquoi les décideurs politiques requièrent des outils pour élaborer, étudier et évaluer des politiques dans un environnement en mutation.

De notre point de vue, le processus de conception de politiques est un ensemble d'essais/erreurs où les hypothèses sont expérimentées sur un modèle réaliste. Cette méthode convient à la simulation multi-agent où les politiques sont créées et éprouvées avant d'être améliorées ou abandonnées en fonction de leurs résultats. L'architecture proposée dans ce papier vise principalement à concevoir des politiques urbaines s'adaptant aux populations, pour des villes tournées vers le futur, que nous appliquons ici au domaine de la mobilité.

1.2 Objectifs

Nous proposons ici un modèle générique pour la conception de politiques urbaines, basée sur une simulation multi-agents individu centrée. Appliquée au domaine de la mobilité pendulaire, les objectifs du modèle sont les suivants : 1) capacité à représenter des usagers et leur utilisation du réseau de transport urbain pour les déplacements domicile-travail et 2) capacité à simuler des politiques de mobilité particulières, en observant leur impact sur l'environnement et les conséquences sur les usagers.

1.3 Contributions

Nous proposons d'abord SmartGov, une approche innovante et générique couplant deux simulations multi-agents. L'une modélise le monde en se basant sur des données géographiques disponibles, garantissant le réalisme de représentation de la ville (section 3.1). L'autre est un modèle de politiques adaptatives représentant la couche décisionnelle et utilisant des agents politiques ayant une perception limitée de leur environnement (section 3.2).

Deuxièmement, un formalisme décrivant la politique urbaine est proposé ainsi que des supports d'interaction permettent aux décideurs politiques d'interagir avec notre modèle suivant différentes méthodes (section 3.3). Le processus de construction entre le décideur et SmartGov pour l'adaptation dynamique des politiques est ensuite détaillé (section 3.4).

Troisièmement, nous explicitons notre modèle sur un exemple (section 2) qui sert de support à l'implémentation du modèle (section 4) et nous montrons que, à l'aide de techniques d'apprentissage par renforcement, une politique urbaine conçue avec notre outil est capable de s'adapter aux objectifs définis, pour une population particulière (section 4.3).

Ce papier démarre avec le cas d'utilisation choisi pour l'élaboration de politique, lié à la tarification dynamique d'emplacements de stationnement en centre ville (section 2). La section suivante introduit notre modèle générique couplant un modèle du monde et un gestionnaire de politiques (section 3). Puis, l'exemple de tarification est implémenté à l'aide de notre modèle et étudié avec les décideurs politiques.

2 Exemple de modélisation

Dans cette partie, nous illustrons le concept principal de notre système sur un exemple simplifié : un décideur a besoin d'élaborer une politique tarifaire d'emplacements de stationnement *on-street* (appelés emplacements dans le reste de l'article). L'observation actuelle est que les usagers passent un certain temps à chercher un emplacement, générant de la pollution et de l'insatisfaction. La motivation sous-jacente est d'assurer une efficacité globale de l'attribution des emplacements et viser ainsi une réduction du trafic aux heures de pointes. En fonction de la finesse attendue par la politique, le décideur fournit le réseau routier ainsi que les informations

sur les emplacements, le type des bâtiments et si possible les matrices origine-destination représentant les trajets domicile/travail (fig. 1).

Ces informations servent à définir l'environnement cible de la politique, ou *périmètre*. Il inclut différentes *structures* représentant les objets de l'environnement, comme les emplacements qui peuvent être modifiés par la future politique. Ainsi on imagine qu'une infrastructure dynamique pourra, en cas de trafic réduit et à l'aide d'un marquage dynamique, transformer une partie de voie en aires de stationnement par exemple, pour augmenter la dimension du parc. Les emplacements comme les bâtiments sont décrits par un ensemble d'informations qualitatives et quantitatives : heures d'ouvertures, prix, nombre de résidents ; dont les valeurs dépendent du type d'élément. Dans notre modèle, ces attributs sont appelés *perceptions* et décrivent des *types* spécifiques de structures.

Une fois la description du *périmètre* achevée, le décideur a besoin de représenter les travailleurs pendulaires, qui sont les principales cibles de la politique. En fonction des réactions des travailleurs, le décideur évalue les *performances* de sa politique et peut augmenter le prix dans certains quartiers et le baisser dans d'autres.

Le réalisme des agents représentant la population est difficilement objectif et des hypothèses sont faites à partir d'études sociales. Dans notre modélisation, les travailleurs sont représentés par des agents considérés comme étant rationnels et utilitaristes. Chaque profil a ses propres préférences, qui peuvent être calibrées par le décideur en fonction du *périmètre* étudié. Une distribution de profils représente une population.

L'environnement et les agents peuvent être modélisés dans une simulation multi-agents individu-centrée. Avant d'évaluer la simulation, le décideur peut fournir ses attentes et débuter une simulation pour trouver les politiques pertinentes relatives à ses objectifs (section 3.3).

Dans notre modélisation, chaque modification de l'environnement effectuée par le décideur représente une *action politique*. Pour notre exemple, une action peut être de modifier la tarification des emplacements d'une zone donnée. Une *politique urbaine* est une séquence d'actions politiques portant des informations sur où et quand appliquer chaque action de la séquence. Comme une *politique urbaine* peut en final se révéler contre-productive, une aide au décideur est nécessaire pour continuellement ajuster les actions afin de s'assurer du succès de la politique.

FIGURE 1 – Agents humains (triangles noirs) cherchant un emplacement dans la ville. Les emplacements disponibles (carrés gris clair) et occupés (carrés gris foncé) ainsi que les lieux de travail (gris foncé) et les résidences (gris clair) sont des *structures* définissant le *périmètre*.

Un autre aspect de la modélisation réaliste et dynamique est celui de la mise à l'échelle. Ainsi rien qu'en considérant l'exemple de la fig. 1 à l'échelle d'une grande ville, l'ensemble des politiques tarifaires à considérer pour chaque emplacement serait déjà très grand pour évaluer toutes les alternatives possibles au niveau global. Une seconde simulation multi-agent est alors adjointe à la première, composée d'agents politiques locaux, qui comme les agents humains, interagissent sur une partie restreinte de l'environnement. Alors que les agents humains se déplacent sur le réseau, les agents politiques définissent les actions à effectuer sur les structures présentes dans leur périmètre.

3 Modèle formel

Cette section décrit notre modèle SmartGov, basé sur des concepts introduits dans la section précédente (structure, périmètre, indicateurs, etc.). Le modèle est générique et permet l'élaboration de politiques urbaines, la possibilité d'effectuer des tests sur un environnement défini, et la validation par une évaluation dynamique de l'impact des politiques sur l'environnement simulé. SmartGov repose sur un couplage dynamique de deux simulations multi-agents : une pour proposer une représentation réaliste de la ville et une pour appliquer les actions politiques. Nous appelons *Couche Usagers* (CU) la couche représentant la simulation de l'environnement et des agents humains. La seconde couche est appelée *Couche du Gestionnaire de Politiques* (CGP) et utilise des techniques d'ap-

prentissage pour prendre des décisions répondant aux attentes du décideur.

3.1 Couche Usagers

Cette partie introduit la *Couche Usagers* qui est un simulateur multi-agents d'un environnement $\mathcal{E}$ et d'une population d'agents humains N_h évoluant dans cet environnement. L'instanciation du modèle générique de la CU pour notre exemple de mobilité urbaine, sera fourni en section 4.1.

L'environnement est ainsi défini par le tuple $\mathcal{E} = \langle \mathcal{S}, T, I, G, N_h \rangle$ où $\mathcal{S}$ est un ensemble fini de structures ; T est un ensemble fini de types de structures ; I est un ensemble fini de perceptions ; G est un réseau défini par $G = \langle V, AR, \omega \rangle$ avec V un ensemble fini de noeuds, $AR \subseteq V \times V$ un ensemble fini de segments et $\omega : AR \rightarrow \mathbb{R}^+$ une fonction de poids ; enfin N_h est un ensemble fini d'agents humains. Nous appelons s_{λ_i} la valeur de la perception i, pour la structure s, définie sur un espace algébrique $\Lambda_i \subseteq \mathbb{R}$. Les structures sont décrites par leur type et un sous-ensemble de perceptions $I' \subseteq I$: une structure $s \in \mathcal{S}$ est donc un couple $s = \langle t, I' \rangle$.

Les agents présents dans cette couche sont une représentation des agents humains dans une ville. Notre modèle de comportement est générique, permettant différentes descriptions d'agents humains et de leurs réactions. L'ensemble des agents humains est $N_h = \{a_h^1, a_h^2, \ldots, a_h^N\}, N \in \mathbb{N}$. Un agent humain a_h^n utilise un automate à états finis non déterministe pour décrire son comportement. Il est ainsi défini par $a_h^n = \langle \Sigma, S, e_0, F, \delta, P^n \rangle$ où $\Sigma = \langle v_p, op, v_t \rangle$ décrit un alphabet où $v_p \in \Lambda$ décrit une valeur perçue dans l'ensemble fini de perceptions disponibles $I_h^n \subseteq I$, op est un ensemble fini d'opérateurs d'égalités et d'inégalités et $v_t \in \Lambda$ est le seuil à atteindre pour valider l'égalité ou l'inégalité ; S est un ensemble fini d'états de l'automate représentant l'agent humain ; $e_0 \in S$ est l'état initial ; $F \in S$ est l'état final ; δ est une fonction de transition entre les états où $\delta = \Sigma \times S \rightarrow P(S) | P(S) \subseteq S$; et $P^n = \left\{ (w_i^n, \sigma_i) \mid i \in I_h^n, w_i^n \in \mathbb{R}^+ \right\}$ est appelé la *personnalité* de l'agent. Le poids w_i^n décrit l'importance attribuée, par un agent, à la perception i. La fonction de score $\sigma_i : \Lambda_i \rightarrow [0; 1]$ définit la réponse de l'agent à la valeur de la perception λ_i. Par exemple, σ_i peut être linéaire $\sigma_i(\lambda_i) = a\lambda_i + b$ ou encore logarithmique $\sigma_i(\lambda_i) = log(1 + \lambda_i)$ pour $\lambda_i \in \Lambda_i$.

L'agent humain a_h^n est utilitariste : il utilise sa personnalité pour attribuer un score aux structures à choisir, à l'aide de sa fonction d'utilité $u^n : s \rightarrow \mathbb{R}^+$ (eq. (1)), ce qui lui permet d'estimer le bénéfice d'une action sur une autre.

$$u^n(s) = \frac{\sum_{i=1}^{q} w_i^n \times \lambda_{s_i}}{\sum_{i=1}^{q} w_i^n}, q = \|I^n\| \qquad (1)$$

À noter que l'utilité de l'agent sera toujours comprise entre [0; 1], quels que soient les poids associés.

Nous avons utilisé l'architecture hiérarchique modulaire de Ferber [3] pour simuler un humain. Chaque agent possède donc des capteurs et des actionneurs ainsi qu'un module de prise de décision avec une fonction d'utilité. Chaque fonction d'utilité est modifiée à la discrétion du décideur.

3.2 Couche du Gestionnaire de Politiques

Cette partie introduit la *Couche du Gestionnaire de Politiques* qui est le deuxième simulateur multi-agents de SmartGov : il comprend une description de son environnement $\mathcal{H}$ ainsi qu'une population d'agents politiques N_p.

L'environnement de cette couche, appelé Gestionnaire de Politiques, est défini par le tuple $\mathcal{H} = \langle I_{\mathcal{H}}, R_{\mathcal{H}}, A, N_p \rangle$ où $I_{\mathcal{H}} \subseteq I$ est un ensemble fini de perceptions de l'environnement de CU ; $R_{\mathcal{H}}$ un ensemble fini de représentations d'environnement construit à partir de l'agrégation de perceptions de $I_{\mathcal{H}}$; A un ensemble fini d'actions disponibles pour le gestionnaire de politiques et N_p un ensemble fini d'agents politiques.

L'ensemble des agents politiques est $N_p = \{a_p^1, a_p^2, \ldots, a_p^M\}$, $M \in \mathbb{N}$. Un agent politique $a_p^m = \langle \mathcal{S}^m, I_p^m, A^m, \rho \rangle$ où $\mathcal{S}^m \subseteq \mathcal{S}$ décrit un ensemble fini de structures de CU où l'agent est en mesure de percevoir $I_p^m \subseteq I_{\mathcal{H}}$ et possède $A^m \subseteq A$ actions disponibles sur les structures $\mathcal{S}^m$; et où $\rho : I_p^m \times A^m \rightarrow \mathcal{S}^m$ est une fonction de décision pour choisir l'action qui correspond à la perception fournie par l'environnement. Les agents politiques sont distribués sur l'environnement : ils sont responsables d'une zone restreinte et appliquent des actions locales pour augmenter les performances locales et globales.

Par analogie avec les agents humains, les agents politiques peuvent percevoir leur environnement

et agir dessus. $I_{\mathcal{H}}$ représente la totalité des capteurs d'un gestionnaire de politique et A représente la totalité de ses actionneurs. Ces capteurs et actionneurs sont fournis à la population d'agents politiques. Les actionneurs du gestionnaire de politiques ne peuvent modifier que la partie commune entre l'environnement $\mathcal{E}$ et le gestionnaire de politiques $\mathcal{H}$, soit l'ensemble $\mathcal{S}_{\mathcal{H}} \subseteq \mathcal{S}$. Nous proposons donc l'ensemble des *perceptions actionnables* I_{act} et l'*action politique* $\alpha \in A$. Une *perception actionnable* $i_{act} \in I_{act}$ est une perception pour laquelle la valeur λ_i peut être changée à n'importe quel moment durant la simulation par la *Couche du Gestionnaire de Politiques*. Une action politique est une fonction qui utilise i_{act} présent dans une structure $s \in \mathcal{S}_{\mathcal{H}}$. L'action politique $\alpha : \Lambda \rightarrow \Lambda$ avec $\alpha(s_{\lambda_i}) = s_{\lambda_i'}$ représente la modification d'une valeur spécifique d'une perception i par l'action α sur la structure s. De manière similaire, la portée des capteurs du gestionnaire de politiques est l'ensemble $I_{\mathcal{H}}$ défini sur les structures $\mathcal{S}_I$. Par conséquent, la couche du gestionnaire de politiques ne peut pas directement changer les paramètres des agents humains mais peut, à travers différentes actions politiques, encourager des changements de comportement. Une politique qui va dans ce sens est appelée *politique incitative*.

3.3 Supports pour l'interaction

Le principal atout de notre système est de permettre aux décideurs de tester une politique urbaine sans avoir à la mettre en oeuvre dans la réalité, avec les moyens coûteux que cela peut engendrer. Notre objectif est ainsi de visualiser directement les conséquences d'une politique et d'en étudier la validité sur la modélisation réaliste de l'environnement qui a été faite.

Dans cette section, nous proposons de décrire les outils mis à disposition du décideur pour interagir avec l'outil de simulation SmartGov. Pour cela, quatre paramètres sont introduits. Le premier paramètre *périmètre environnemental* $\psi = \langle G, I, N, \mathcal{S} \rangle$, décrit le périmètre sur lequel la politique va être appliquée avec un graphe, les perceptions disponibles, les structures et où $N = \|N_h\|$ définit le nombre d'agents humains de la population. Le second paramètre, le *domaine d'action* φ définit la probabilité que l'action α soit utilisée dans l'environnement $\varphi : A \rightarrow [0; 1]$, où le domaine d'action appliqué à la structure $s \in S$ est $\varphi(s) = \{(\alpha_i, p(\alpha_i)) | \alpha_i \in A, p(\alpha_i) \in [0; 1]\}$ où α_i décrit une action et $p(\alpha_i)$ la probabilité de faire cette action. En utilisant ces probabilités, le décideur peut choisir d'autoriser ou d'interdire certains actions sur l'environnement. Le troisième paramètre, les *indicateurs de performances* ϕ décrivent des fonctions à maximiser ou à minimiser et permet de décrire une fonction objectif. Le quatrième paramètre est l'*horizon de temps* H sur lequel les politiques seront appliquées.

Pour faciliter la compréhension du processus d'interaction, nous proposons les définitions suivantes relatives aux politiques urbaines :

Définition 1 *Une* politique urbaine *représente l'application d'un ensemble ordonné d'actions sur l'environnement pendant un certain temps afin de satisfaire les objectifs fournis.*

Comme mentionné précédemment, les décideurs peuvent co-construire des politiques publiques avec l'outil de simulation, qu'ils utilisent avec des attentes différentes, en fournissant une entrée E : (i) simple analyse de l'environnement ζ ; (ii) atteindre des objectifs spécifiques O ; (iii) tester une politique publique $\mathcal{P}$.

Définition 2 *Une* analyse d'environnement $\zeta = \langle \psi, \phi, H \rangle$ *est définie par un périmètre environnemental* ψ*, des indicateurs de performances* ϕ *à évaluer et un horizon de temps* H*.*

Définition 3 *Un* objectif politique $O = \langle \phi, \varphi, \psi, H \rangle$*, fourni par le décideur, est défini par des indicateurs de performances* ϕ *à optimiser ;* φ *le domaine d'action choisi par le décideur ;* ψ *le périmètre environnemental sur lequel la politique va être appliquée et* H *représente la date au plus tard à laquelle* O *devrait être satisfait.*

Définition 4 *Une* politique publique $\mathcal{P} = \langle \mathcal{F}, O \rangle$ *est définie par* $\mathcal{F}$ *un ensemble fini de fonctions d'actions politiques, tel que* $f_i : R_i \rightarrow A_i | i \in N_p, f_i \in \mathcal{F}, R_i \subseteq R, A_i \subseteq A$ *et* $f_i(r_i) = \alpha_i$ *représente l'action politique courante à appliquer quand la représentation locale des agents politiques* a_p^i *est* r_i *;* O *les objectifs politiques spécifiés par le décideur.*

Une analyse d'environnement ζ crée un état initial à partir des structures spécifiés. ζ ne permet pas de tester des politiques mais uniquement d'observer que le comportement des agents et l'environnement sont cohérents. L'objectif d'une analyse d'environnement est de permettre la visualisation et l'évaluation de l'environnement et des agents humains à partir des données fournies, par exemple dans un but de validation du réalisme de la simulation. Fournir au simulateur

un objectif politique O crée un état initial à partir des actions, structures et des perceptions disponibles, avec des performances que les agents politiques doivent maximiser. Cette fois, la simulation applique des actions politiques parmi celles possible, et évalue la performance de ces actions. Enfin le décideur peut proposer une politique publique $\mathcal{P}$, avec l'objectif d'en évaluer les performances. SmartGov produit toujours une politique publique $\mathcal{P}$ quelle que soit l'entrée E fournie.

3.4 Processus de SmartGov

La première étape consiste à construire les deux simulateurs à partir de l'entrée E du décideur. ζ, O et $\mathcal{P}$ permettent de créer une instance d'environnement $\mathcal{E}$ alors que O et $\mathcal{P}$ servent à créer une instance de gestionnaire de politiques $\mathcal{H}$. $\mathcal{E}$ structure l'environnement et construit des agents humains réalistes. $\mathcal{H}$ construit des agents politiques qu'il distribue sur les structures de l'environnement.

La seconde étape est le fonctionnement du couplage entre les deux simulateurs (fig. 2) : la population N_h interagit avec les structures $\mathcal{S}$ défini par Σ et provoque des changements microscopiques. La population N_p observe $r_t \in R_{\mathcal{H}}$ à un instant t correspondant à une agrégation de $I_{\mathcal{H}}$ des structures modifiées par les agents humains (ex. : se garer sur un emplacement). a_p^m effectue une action α, modifiant les structures $\mathcal{S}^m$ de l'environnement (augmentation du prix des emplacements par exemple). En retour, N_h observe les différences et modifie son comportement vis-à-vis de cette modification. Par la suite, des modifications du niveau microscopique seront observées (occupation plus faible des emplacements).

La troisième étape est l'évaluation des actions effectuées par les agents politiques. À n'importe quel moment, $\mathcal{H}$ peut arrêter la simulation et évaluer les performances. Le calcul des performances se fait à l'aide de la fonction objectif définie par les indicateurs ϕ lorsque le décideur fourni O ou $\mathcal{P}$. En fonction des performances, le système applique des actions particulières afin d'améliorer les résultats et/ou rendre la simulation plus efficace. La comparaison entre les performances et les objectifs permet de déterminer la pertinence de la politique actuelle.

Quand les agents politiques appliquent une action, une nouvelle simulation démarre avec les valeur des structures mises à jour. Deux autres approches peuvent être identifiées pour représenter la réaction des agents humains : (i) une réaction sans prendre en compte le temps, c'est-à-dire sans avoir à modéliser la phase d'adaptation. Dans ce cas, quand une nouvelle politique est implémentée, nous supposons que les agents humains n'ont aucune mémoire des actions précédentes et qu'ils agissent comme si la situation était celle à laquelle ils sont habitués. (ii) une réaction en prenant en compte le temps d'adaptation. Dans ce cas, quand l'action est appliquée, les agents mettent du temps pour s'adapter et changer leur comportement. Dans notre contexte, pour évaluer l'efficacité de la politique, nous sommes intéressés par l'état stable obtenu après la période d'adaptation. Nous avons donc choisi d'utiliser l'approche (i).

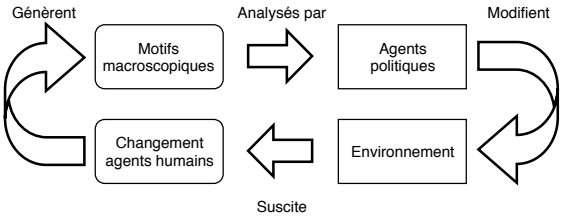

FIGURE 2 – Fonctionnement du couplage entre les deux simulateurs multi-agents. Les agents humains agissent sur l'environnement modélisé et produisent des résultats macroscopiques qui sont analysés par la CGP à δt. Cette dernière décide ensuite d'une ou plusieurs actions à effectuer sur l'environnement de la CU.

4 Implémentation

Cette section décrit l'implémentation de l'exemple proposé (section 2) avec notre modèle (section 3) et où le décideur souhaite utiliser SmartGov pour évaluer une politique de tarification dynamique des places de stationnement en centre ville. Dans ce contexte, CU décrit le comportement des usagers et CGP décrit comment les agents politiques modifient l'environnement pour satisfaire les objectifs du décideur. Notre outil de simulation apprend quand changer les tarifs en utilisant un apprentissage par renforcement. L'implémentation est utilisée pour évaluer l'impact global de la politique de tarification en centre ville.

4.1 Description

L'objectif est de fournir une implémentation du modèle pour décrire le fonctionnement de SmartGov. L'environnement $\mathcal{E}$ est dé-

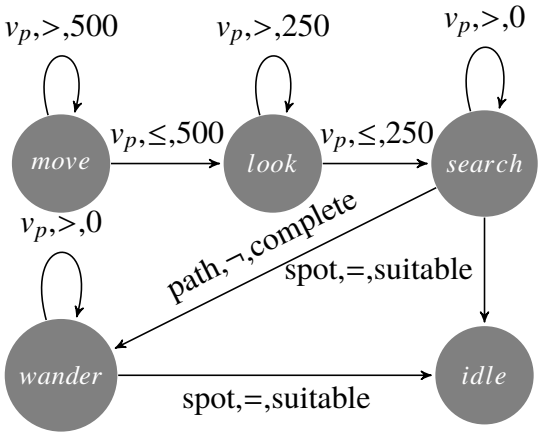

FIGURE 3 – Cet automate à états finis est utilisé pour la recherche d'emplacements par N_h. L'état initial est *move* et l'état final est *idle*, atteint lorsque l'agent atteint un emplacement satisfaisant.

crit par : l'ensemble des structures comprenant les emplacements $\mathcal{S}_{emp}$ et les bâtiments $\mathcal{S}_{bat}$; T est l'ensemble des types des structures tel que $T_{\mathcal{S}_{emp}} = \{\text{off-street}, \text{on-street}\}$ et $T_{\mathcal{S}_{bat}} = \{\text{residential}, \text{leisure}, \text{work-office}\}$; I est l'ensemble des perceptions considérées par l'agent humain a_h^n. Ici, trois perceptions sont considérées : $i_{d(s,s')}$ la distance entre deux structures, $i_{p(s)}, s \in \mathcal{S}_{spots}$ le tarif d'un emplacement et i_r le temps de recherche d'un emplacement satisfaisant. G est un graphe orienté représentant le réseau routier de la ville sur lequel la population d'agents humains N_h va se déplacer.

a_h^n utilise un automate à états finis déterministe lié au scénario (fig. 3) tel que l'alphabet Σ est basé sur la distance entre la position actuelle de l'agent v_p et son objectif v_t et les opérateurs $op = \{=, \leq, \neg, >\}$ décrivent les conditions à satisfaire pour passer de l'état s à s'; $s, s' \in S$. Par exemple, l'instruction suivante : $v_p, \leq, 500$ est vraie si l'agent est à 500 mètres ou moins de sa destination. L'ensemble des états de l'automate est décrit par $S = \{\text{move}, \text{look}, \text{search}, \text{wander}, \text{idle}\}$; l'état initial est $e_0 = \{\text{move}\}$ et l'état final $F = \{\text{idle}\}$ est atteint quand l'agent trouve un emplacement satisfaisant et qu'il s'y gare.

Afin d'expérimenter les politiques urbaines sur différentes populations et comme évoqué précédemment, nous utilisons des profils de travailleurs. En utilisant la fonction d'utilité (eq. (1)), nous proposons cinq profils aux comportements distincts qui permettent d'avoir une population hétérogène. Les intervalles utilisés pour décrire la réaction des agents dépendent des perceptions de leur environnement. L'utilité est égale à 1 lorsque la valeur perçue est dans le meilleur intervalle, à 0 lorsque la valeur perçue est dans le mauvais intervalle et elle évolue linéairement entre les deux intervalles.

a_h^n perçoit l'environnement à partir des trois perceptions introduites précédemment : $i_{d(s,s')}$, $i_{p(s)}$ et i_r. Ces perceptions varient en cours de la simulation : la distance entre l'emplacement et l'objectif de l'agent $\lambda_{i_{d(s,s')}} \in \mathbb{R}^+, s \in \mathcal{S}_{emp}, s' \in \mathcal{S}_{bat}$ entre 0 et 1000 mètres ; le prix actuel de l'emplacement $\lambda_{i_{p(s)}} \in \mathbb{Q}^{+*}$ entre 0.5$ et 4$ et le temps de recherche d'un emplacement satisfaisant $\lambda_r \in \mathbb{N}$ entre 0 et 600 secondes. Soit $\mathcal{S}'_{emp}$ l'ensemble des structures à évaluer, a_h^n va choisir un emplacement $s' \in \mathcal{S}'_{emp}$ qui maximisera sa fonction d'utilité.

En plus de N_h représentant les travailleurs, nous utilisons une population d'agents politiques N_p qui gère l'adaptation locale de la politique à partir des ses perceptions locales. Ils sont distribués sur différents fronts de rue, chacun possédant au moins un emplacement. a_p^m est décrit par : un sous-ensemble d'emplacements $\mathcal{S}_{emp}^m \subseteq \mathcal{S}_{emp}$ tel que $T_{\mathcal{S}_{emp}^m} = \{\text{on-street}\}$; ses perceptions $I_p^m = \{i_{\text{occ}}, i_{\text{pri}}\}$, où i_{occ} est le nombre d'emplacements occupés dans le front de rue et i_{pri} le prix des emplacements du front de rue (que l'on suppose uniforme) ; et ses actions disponibles $A^m = \{\alpha(\nearrow), \alpha(\searrow), \alpha(=)\}$ où $\alpha(\nearrow)$ augmente les prix, $\alpha(\searrow)$ diminue les prix et $\alpha(=)$ ne fait rien. Enfin, la fonction de décision ρ est basée sur une technique d'apprentissage par renforcement pour automatiquement adapter les actions de la politique pour satisfaire les objectifs du décideur. La régulation tarifaire de notre exemple suit donc une approche décentralisée dans laquelle chaque a_p^m gère son front de rue et applique des actions localement.

4.2 Expérimentation

Le modèle est implémenté sur la plateforme Repast Simphony version 2.5 [5]. Le réseau routier est généré à partir des données de OSM (Open Street Map) et reconstruit avec un outil d'extraction spécialement conçu pour SmartGov. Les informations sur les infrastructures de la ville, telles que les bâtiments et les lignes de métro par exemple, sont également extraites d'OSM. Les bâtiments sont utilisés pour attribuer des lieux de travail aux agents. Le décideur extrait le

TABLE 1 – Les cinq profils d'agents humains et les poids de leur fonction d'utilité

Profil	Distance (mètres)			Prix ($)			Temps de recherche (secondes)		
	Meilleur	Pire	Poids	Meilleur	Pire	Poids	Meilleur	Pire	Poids
Normal	[0 :250]	[500 :1000]	1	[0 :2.5]	[3.5 :4]	1	[600]	[0 :300]	1
Économe	[0 :250]	[500 :1000]	0.5	[0 :1.5]	[2.5 :4]	1	[600]	[0 :300]	0.5
Écolo	[0 :500]	[1000]	1	[0 :2.5]	[4]	0.5	[600]	[0 :300]	0.5
Impatient	[0 :250]	[350 :1000]	0.2	[0 :2.5]	[4]	0.2	[150 :600]	[0 :100]	1
En retard	[0 :100]	[200 :1000]	1	[0 :4]	[-]	0.2	[150 :600]	[0 :100]	0.2

périmètre environnemental et le fournit à l'outil de simulation. Toutes ces données donnent une description réaliste du modèle de la simulation. Dans notre cas, la position des emplacements provient de précédents travaux sur la tarification dynamique [9], nous permettant de simuler un quartier réel de Los Angeles (fig. 1). Trois populations (table 2) sont instanciées combinant les cinq profils (table 1). Le quartier ciblé est composé de 32 fronts de rues qui ont entre 1 et 21 emplacements disponibles, pour un total de 243 emplacements. Le scénario actuel utilise 500 agents humains et 32 agents politiques. L'origine et la destination des agents sont tirées aléatoirement sur les bâtiments adéquats, et les agents ne peuvent pas sortir de la simulation en cours.

Dans cet exemple, nous considérons que le décideur souhaite utiliser SmartGov pour évaluer sa politique, avec l'objectif de maximiser le nombre d'agents stationnés et de minimiser le temps de recherche. Pour chaque front de rue, a_p^m observe l'occupation ainsi que la moyenne, pour chaque a_h^n présent dans sa zone, du temps de recherche et de la distance entre les emplacements et la destination cible. Dans cette implémentation, les récompenses sont basées sur l'occupation et le temps de recherche en concordance avec les objectifs du décideur.

Un algorithme de bandit à n-bras est utilisé comme technique d'exploration pour l'apprentissage par renforcement des agents politiques. Ici, chaque agent politique possède deux bras : un pour décroître les tarifs, et un autre pour les augmenter. La modification du prix permet à a_p^m de réguler l'occupation et le temps de recherche. Il est ainsi nécessaire de choisir un bras à actionner et observer les résultats, sans savoir ce que les autres bras auraient donné. Plusieurs algorithmes existent pour l'implémentation du bandit à n-bras mais nous choisissons l'algorithme *Upper Confidence Bound (UCB)* [1] qui garantit un bon compromis entre exploitation du meilleur bras à utiliser et exploration des autres bras disponibles. L'initialisation d'un UCB requiert d'essayer chaque bras au moins une fois pour voir quel bras maximise $\overline{x_j} + \sqrt{\frac{2\ln(n)}{n_j}}$, où $\overline{x_j}$ est la récompense moyenne obtenue pour le bras j ; n_j est le nombre de fois où le bras j a été actionné et n est le nombre total d'essais.

Les expérimentations sont conduites en utilisant un scénario illustrant les interactions possibles entre le décideur et notre système. En premier lieu, le décideur demande une analyse d'environnement ζ pour évaluer le comportement initial du modèle dans sa configuration actuelle (fig. 4). Le décideur observe ainsi que le temps de recherche et la distance moyenne pour chaque profil réagissent différemment à l'évolution des prix. À partir de ces observations, le décideur propose un objectif à atteindre qui tienne compte du prix. Comme introduit précédemment, le décideur souhaite maximiser le nombre d'agents stationnés et limiter le temps de recherche, donc la politique attendue doit proposer un compromis entre ces critères. Le souhait du décideur se traduit par un objectif O décrivant le domaine d'action φ et donc les actions disponibles $A = \{\alpha(\nearrow), \alpha(\searrow), \alpha(=)\}$s, ainsi que la fonction de récompense, pour une instance de gestionnaire de politiques $\mathcal{H}$. Pour chaque itération du cycle de simulation, les agents politiques peuvent choisir d'augmenter, de diminuer ou de conserver les tarifs actuels. Une fois les objectifs fixés, le décideur démarre une simulation.

4.3 Résultats

L'évaluation de la pertinence d'une politique publique repose sur les observations des performances globales après avoir appliqué plusieurs actions politiques locales sur un horizon fini. Les résultats de la fig. 5 montrent les deux indicateurs de performance globaux proposés : le temps de

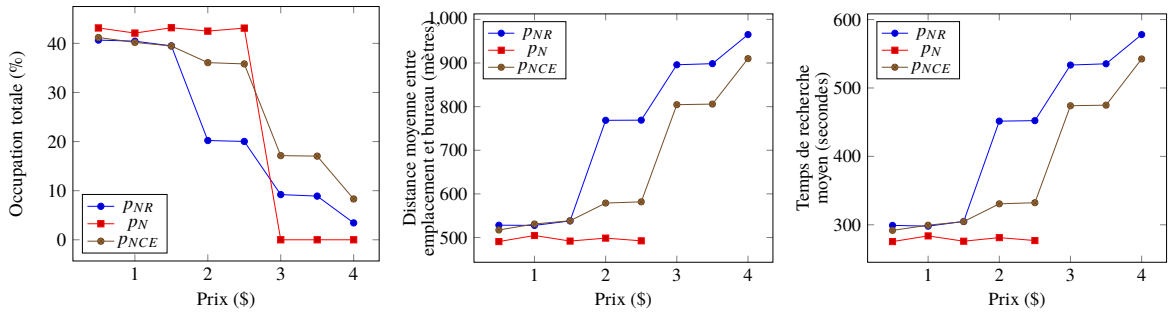

FIGURE 4 – Résultats d'une analyse d'environnement ζ pour les trois populations de table 2

TABLE 2 – Trois populations créées à l'aide des profils de table 1 pour avoir des comportements hétérogènes

	Profils (%)				
Pop	Nor	Écol	Econ	Imp	Ret
p_{NR}	60	10	10	0	20
p_{NCE}	50	20	20	5	5
p_N	100	0	0	0	0

recherche moyen pour trouver un emplacement (en secondes), et le pourcentage d'agents stationnés. Nous observons que, après avoir effectué un certain nombre d'actions, le nombre d'agents stationnés augmente et le temps de recherche moyen pour trouver un emplacement diminue. Chaque action politique correspond à une modification du tarif, effectuée par un agent politique sur son front de rue, en utilisant la technique d'apprentissage précédemment décrite. La séquence d'actions qui mène à l'amélioration des performances définit ainsi une politique fonctionnelle, fournissant aux décideurs des informations sur les structures à modifier pour satisfaire leurs objectifs dans la réalité. Les décideurs peuvent effectuer une analyse plus poussée sur la robustesse de la politique en ré-appliquant la même séquence ou en l'appliquant à un autre contexte, sur une autre population, ou sur une autre tranche horaire. Ces résultats sont une illustration du processus d'interaction du décideur avec notre système.

4.4 Discussion

Ce papier est une première étape vers la co-conception de politiques, illustrée avec un exemple de mobilité urbaine. Notre modèle représente actuellement une situation dans laquelle les emplacements sont conservés pour la journée entière, ce qui est cohérent pour des travailleurs

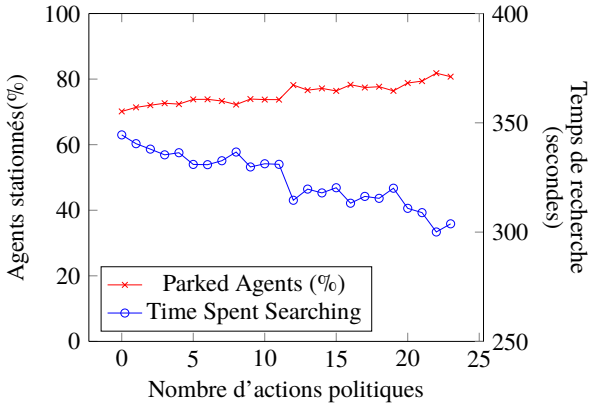

FIGURE 5 – Impact d'une politique sur le pourcentage d'agents stationnés (—×—) et sur leur temps de recherche d'emplacements (—○—)

mais devrait être modifié pour des centres commerciaux ou autre lieux de loisirs où les gens stationnés sur les emplacements changent régulièrement au cours de la journée. Une extension pourrait être d'avoir des politiques par tranche horaire, ce qui signifie que la décision de l'agent politique devrait être dépendante de l'heure où le tarif est appliqué. Il existe des améliorations du réalisme de la simulation comme la prise en compte du transport multi-modal, et des améliorations sur l'algorithme de recherche d'emplacements.

Un autre point concerne la fonction de récompense qui impacte, comme dans beaucoup de problèmes de renforcement, la qualité des résultats. Dans notre modèle, la fonction de récompense dépend des objectifs des décideurs politiques. Ainsi, si le décideur propose des objectifs aberrants, le modèle ne sera pas en mesure de produire une politique pertinente. Un avantage intéressant de notre système est que les agents politiques n'ont pas besoin d'informations sur la population d'agents humains pour appliquer des actions politiques augmentant leur récompense.

En effet avec l'utilisation du bandit à n-bras, les détails sur une population sont appris au cours de l'apprentissage, et l'agent politique peut donc s'adapter aux populations. Les perspectives envisagées portent sur l'amélioration de la fonction de récompense, de la technique d'apprentissage et l'expérimentation de scénarios différents. Il est important de noter que ce modèle est générique et vise à concevoir des politiques urbaines, ça n'est pas un modèle ciblé pour la recherche de places de stationnement en ville.

5 Conclusion

Dans ce papier, nous avons présenté une manière innovante de concevoir et d'évaluer des politiques urbaines. Nous avons décrit SmartGov, une architecture générique couplant une simulation du monde et un gestionnaire de politiques utilisant des simulations multi-agents. Un formalisme a été proposé pour représenter des politiques urbaines génériques et un processus interactif est proposé, permettant aux décideurs de précisément comprendre la conception de politiques par l'outil et ses résultats, facilitant le calibrage des représentations utilisées et l'amélioration progressive des objectifs à définir. Afin de réduire l'explosion combinatoire des états du système, notre approche utilise une population d'agents politiques appliquant des actions locales à l'aide de techniques d'apprentissages par renforcement avec l'algorithme du bandit à n-bras. Ce modèle est instancié sur un quartier d'une ville réelle, pour une politique de tarification dynamique d'emplacements, et propose des résultats qui suggèrent des modifications aux décideurs, adaptées au contexte et à l'environnement.

En l'état actuel, les agents politiques sont distribués dans l'environnement en utilisant une méthode arbitraire, attribuant un agent local à chaque front de rue. Une première perspective est d'envisager la division intelligente et dynamique du périmètre considéré, avec là aussi un apprentissage permettant d'améliorer la performance des agents politiques. Une autre piste est de laisser les agents politiques identifier les relations entre actions et perceptions par eux-mêmes, ce qui ne nécessite pas de connaissance particulière, permettant aux agents d'avoir une prise de décision autonome. Les résultats de notre première expérimentation montre d'intéressantes améliorations pour une politique de mobilité urbaine et d'autres opportunités sont déjà considérées.

6 Remerciements

Ce projet est financé par la Région Auvergne-Rhône-Alpes (Contrat ADR ARC7). Il est le fruit d'une collaboration entre le LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information) à Lyon et NLE (NAVER LABS Europe), anciennement XRCE (Xerox Research Center Europe) à Meylan.

Références

[1] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47 :235–256, 2002.

[2] Steven C. Bankes. Tools and techniques for developing policies for complex and uncertain systems. In *Proceedings of the National Academy of Sciences*, volume 99, pages 7199–7200, 2002.

[3] Jacques Ferber. *Multi-Agent System : An Introduction to Distributed Artificial Intelligence*, volume 1. Addison-Wesley, 1999.

[4] Robert J. Lempert, Steven W. Popper, and Steven C. Bankes. *Shaping the Next One Hundred Years : New Methods for Quantitative, Long-Term Policy Analysis*. RAND, 2003.

[5] Michael J. North, Nicholson T. Collier, Jonathan Ozik, Eric R. Tatara, Charles M. Macal, Mark Bragen, and Pam Sydelko. Complex adaptive systems modeling with repast simphony. *Complex Adaptative Systems Modeling*, 2013.

[6] A. J. Scott and M. Storper. The nature of cities : the scope and limits of urban theory. *International Journal of Urban and Regional Research*, 39(1) :1–15, 2015.

[7] Rebecca Sutton. *The policy process : an overview*. London : ODI Publ., 1999.

[8] Alexey Voinov, Nagesh Kolagani, Michael Keith McCall, Pierre Glynn, Marit Ellen Kragt, Frank Ostermann, Suzanne Alise Pierce, and Palaniappan Ramu. Modelling with stakeholders - next generation. *Environmental Modelling and Software*, 2016.

[9] Onno Zoeter, Christopher Dance, Stéphane Clinchant, and Jean-Marc Andreoli. New algorithms for parking demand management and a city scale deployment. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1819–1828, 2014.