



**HAL**  
open science

# Modèles d'auto-organisation pour l'émergence de formes urbaines à partir de comportements individuels à Bogota

Javier Gil Quijano

## ► To cite this version:

Javier Gil Quijano. Modèles d'auto-organisation pour l'émergence de formes urbaines à partir de comportements individuels à Bogota. Autre [cs.OH]. Université Pierre et Marie Curie - Paris VI, 2007. Français. NNT: . tel-00270015

**HAL Id: tel-00270015**

**<https://theses.hal.science/tel-00270015>**

Submitted on 3 Apr 2008

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

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

# THESE de DOCTORAT de L'UNIVERSITE PARIS 6

Spécialité

**Informatique**

Présentée par

**Javier GIL QUIJANO**

Pour obtenir le grade de

**DOCTEUR de l'UNIVERSITÉ PARIS 6**

Sujet de la thèse :

**Modèles d'auto-organisation pour l'émergence de formes urbaines à partir de comportements individuels à Bogota**

Soutenue le 25 septembre 2007

devant le jury composé de :

M.	Alexis DROGOUL	(Directeur de Recherche, IRD)	Directeur de thèse
M.	René MANDIAU	(Professeur U. Valenciennes et du Hainaut-Cambrésis - LAMIH)	Rapporteur
M.	Jean-Daniel ZUCKER	(Directeur de Recherche, IRD)	Rapporteur
M.	Patrick GALLINARI	(Professeur, Directeur LIP6)	Examineur
Mme.	Marie PIRON	(Chargé de recherche, IRD)	Examineur
Mme.	Cécile TANNIER	(Chargé de recherche, CNRS)	Examineur

# Introduction

## **1. Introduction générale**

En sciences sociales, il existe deux grands types d'approche pour rendre compte des phénomènes sociaux. D'un côté, les approches qui accordent un rôle important dans la détermination des phénomènes sociaux aux structures sociales (école de Chicago, Bourdieu) et d'un autre côté, l'individualisme méthodologique [Weber 1913, 1965 ; Schumpeter 1954 ; Boudon, 1977]. L'individualisme méthodologique affirme que tous les phénomènes socio-économiques peuvent être expliqués par les comportements des individus, leurs dispositions, croyances, ressources et relations : « l'individu est l'atome logique d'analyse » [Boudon, 1977]. Le phénomène social est créé à partir de l'agrégation des actions individuelles (« effets de composition »). Boudon met en évidence « l'effet pervers » des phénomènes de composition en montrant que l'addition d'actions individuelles rationnelles peut produire des effets inattendus et contraires aux intentions de chacun. L'individualisme méthodologique présente une limite majeure, qui vient de sa définition même, dans la mesure où les actions des individus sont fonction de leurs croyances, dispositions et ressources. Or les croyances, dispositions et ressources sont produits par la société. Cette approche ne permet pas de rendre compte de ces relations circulaires.

Dans le premier type d'approches, on trouve notamment l'approche de l'école de Chicago et celle de Pierre Bourdieu. Pour les sociologues de l'école de Chicago (Park, Burgess & Mackensie, 1925 ; Burgess & Park 1926 ; Park 1929 ; Wirth 1938 ), l'unité élémentaire dans l'étude des sciences sociales est l'interaction : « la société est faite d'interactions ». Dans ce cadre a été développé l'écologie urbaine [Park, Burgess & Mackensie, 1925] qui cherche à rendre compte du rapport de plusieurs populations sur un même territoire : « ...la sociologie urbaine est conduite à voir la ville non seulement comme mosaïque de territoires, mais aussi comme agencement de populations d'origines différentes dans un même milieu et dans un même système d'activités ... » [Grafmeyer & Joseph, 1990]. L'écologie urbaine montre que la compétition spatiale entre groupes de populations différentes est le moteur explicatif des mobilités résidentielles. L'espace est un milieu dans lequel l'activité d'adaptation et de coopération des individus ou des collectifs trouve ses ressources. De cette manière, l'écologie urbaine met en relation l'individu et le groupe social ou population à laquelle il appartient.

Cette appartenance définit ses capacités mais aussi ses limites d'appropriation et modification de l'espace.

Pierre Bourdieu, quant à lui, établit un lien entre la socialisation et les actions des individus [Bourdieu, 1972 ;1980]. Ce lien est établi par des structures sociales (*l'habitus*) constituées par l'ensemble des dispositions, schèmes d'action ou de perception que l'individu acquiert à travers son expérience sociale. La socialisation est un processus qui permet aux individus d'acquérir des manières durables de penser, sentir et agir. Pour Bourdieu, *l'habitus* est structuré et prédisposé à fonctionner de manière structurante : structuré parce qu'il est formé par la socialisation, mais aussi structurant parce qu'il détermine les pratiques futures des individus, un ensemble de pratiques nouvelles adaptées au monde social dans lequel il se trouve.

Une approche intermédiaire entre ces deux tendances est la théorie des *systèmes complexes*, originalement développée au sein des sciences physiques et de la chimie et qui est au croisement de la cybernétique (Wiener, 1950), de la théorie des systèmes (Bertalanffy 1968 ; Prigogine & Stengers, 1979 ) et de la théorie de l'information (Shannon, 1948). Cette approche a été introduite ensuite dans les sciences sociales (Pumain, 1989 ; Morin 1990 ; Dauphine 2003). Un système fait référence à un assemblage d'éléments fonctionnant de manière unitaire et en interaction permanente. Un système est dit complexe lorsqu'il est composé d'un grand nombre d'éléments tous liés entre eux car il existe un nombre important d'interactions entre les éléments à différents niveaux. Dans cette théorie, on propose l'idée que des processus dynamiques complexes peuvent également résulter d'interactions simples. *L'émergence* est l'une des principales notions issues de la théorie des systèmes complexes. Dans un phénomène émergent, l'application de quelques règles ou lois à un niveau local et parmi de nombreuses entités en interactions vont produire à un niveau agrégé des structures repérables et ordonnées. En retour, les structures formées au niveau agrégé ont une influence sur l'état futur des entités : il existe des *boucles de rétroaction*, c'est-à-dire que l'état d'une entité a une influence sur son état futur *via* l'état des autres entités ou des structures agrégées dont l'entité a participé à la formation. La dynamique de tels systèmes n'est pas gouvernée mais dirigée au niveau des composants élémentaires du système, en interactions sur et avec leur environnement social et spatial. L'approche par les systèmes complexes propose alors d'examiner les mécanismes responsables de ces dynamiques en mettant l'accent sur les liens dynamiques qui unissent ces entités entre elles et entre les différents niveaux d'organisation du système. Transposée aux sciences sociales, la théorie des systèmes complexes, grâce à la notion des boucles d'interaction, d'une part, permet de dépasser la limite de l'individualisme

méthodologique que nous avons exposé auparavant et d'autre part, tient compte de l'importance des structures sociales dans la détermination d'un phénomène social.

### **1.1. La ville un système complexe façonné par les mobilités résidentielles**

La ville peut être considérée comme un système complexe, en raison du grand nombre de composants et d'acteurs situés à différents niveaux, ainsi que des multiples interactions entre acteurs et entre niveaux. En effet, plusieurs types d'éléments participent à l'organisation du système urbain [Paelinck, 1972] : les actions des « décideurs » politiques, économiques ou culturels, les comportements des résidents tout au long de leur vie, et enfin les structures spatiales elles-mêmes et les contraintes qu'elles imposent aux activités humaines. Transposée aux systèmes urbains, la théorie des systèmes complexes suggère que, pour expliquer leur structuration, il n'y a pas une finalité politique ou économique explicite qu'il serait possible de décrire du point de vue d'une fonction à optimiser. La structure est plutôt le produit involontaire d'interactions multiples entre des acteurs très nombreux. C'est en dépit de la diversité de la personnalité des acteurs, de leurs motivations et de leurs actions, que l'agrégation de leurs comportements produit des régularités, repérables en vertu de l'évolution d'une ville. Ces actions «microscopiques» sont en réalité effectuées sous une contrainte forte. Les acteurs individuels et collectifs qui composent le milieu social d'une ville ne sont pas indépendants mais connectés, informés mutuellement de leurs actions. [Pumain, 1989].

Les études et les travaux sur les dynamiques urbaines ont mis, ces dernières années, l'accent sur le rôle des mobilités spatiales dans les recompositions des territoires métropolitains aussi bien du nord comme du sud (Lelièvre & Lévy-Vroelant, 1992 ; Dureau et al., 2000 ; Bonnet et Desjeux, 2000 ; Levy et Dureau, 2002). La notion de recomposition permet de considérer tout à la fois la permanence et le changement des différents territoires, des populations et des modes de vie qui constituent la ville. Autrement dit, elle permet de prendre en compte les composantes de la ville d'hier et d'identifier les formes urbaines contemporaines se juxtaposant ou se superposant.

La ville apparaît ainsi comme un ensemble d'éléments en interaction qu'il convient d'appréhender au travers de leurs caractéristiques, comme les processus de mobilités résidentielles intra-urbaines, et des relations qui les associent à l'espace urbain. La mobilité

résidentielle est un des moteurs essentiels de la modification de la configuration spatiale d'une ville (densification, mutation, extension spatiale, fusion de sites de peuplement, etc.). Dans le cadre de l'écologie urbaine, la mobilité résidentielle est le résultat de la compétition des groupes sociaux ou des populations différentes et contribue à « à la mise en forme de l'espace » [Isnard et al., 1981] : la mobilité des individus ou des unités collectives dont ils font partie constitue un facteur important de modification de l'organisation. Les comportements pris en compte sont les déplacements d'un lieu à un autre qui provoquent une constante recomposition de l'organisation.

## **1.2. La recherche en systèmes multi-agents**

Les systèmes multi-agents permettent de simuler des systèmes complexes notamment par la prise en compte des interactions individuelles qui sont les moteurs de l'émergence de formes à un niveau collectif. Ils offrent la possibilité d'intégrer dans une même simulation des entités situées à différents niveaux de représentation, notamment des individus et des structures agrégées construites à partir de typologies, d'indicateurs, ou de comportements fondés sur des règles symboliques. A partir de la fin des années 1990, ils ont été très utilisés en géographie (Benenson 1998; Portugali et al. 1997; Sanders et al. 1997 ; Holm & Sanders, 2007 ; Daudé, 2002) et tout particulièrement en modélisation des dynamiques urbaines. Ils ont été utilisés pour reproduire différents phénomènes spatiaux : habitat compact ou dispersé [Kohler et al., 2000], hiérarchie du peuplement [Batty, 2001], formes de ségrégation spatiale (Portugali et al. 1997, Torrens 2001), formes d'expansion urbaine [Barros & Alves, 2003].

La simulation par systèmes multi-agents est fondée sur l'idée qu'il est possible de représenter le comportement d'un individu ou d'un groupe social, par un processus informatique appelé agent qui dispose de sa propre autonomie opératoire. Un agent est capable de percevoir son environnement, de réagir à ses transformations, mais également de communiquer et d'interagir avec d'autres agents, son comportement et ses capacités d'interaction pouvant être spécifiés de façon explicite pour chacune des phases de son évolution.

En faisant interagir un ensemble d'agents, on peut simuler des "mondes artificiels" sur lesquels il est possible de conduire de façon systématique et contrôlée toute une variété

d'expérimentations. Le modélisateur emploie ainsi une simulation multi-agents comme s'il s'agissait d'un laboratoire virtuel :

- en ajoutant, supprimant ou modifiant des agents,
- en intervenant sur leurs connaissances ou leur comportement,
- en faisant varier les conditions environnementales,
- en utilisant les mêmes outils d'analyse de résultats expérimentaux que dans la réalité.

Chaque agent peut être suivi à tout moment dans son évolution et avec le degré de finesse désiré. On exploite alors les capacités des ordinateurs pour traiter les données obtenues, les agréger et les analyser de manière statistique afin de vérifier les hypothèses émises. Il devient envisageable de combiner différents niveaux d'analyse au sein d'un même modèle, et d'en évaluer la pertinence. Dans une optique de modélisation de systèmes complexes, les systèmes multi-agents permettent d'implémenter les interactions circulaires entre les individus ou éléments microscopiques et les structures de niveau supérieur comme des groupes sociaux. On peut donner à ces dernières un degré d'autonomie important au point qu'elles peuvent devenir les entités principales de modélisation.

### **1.3. Application sur Bogotá**

Dans ce travail, nous nous intéressons à l'impact des mobilités résidentielles sur les recompositions urbaines de Bogotá, capitale de la Colombie. A l'image de la plupart des villes latino-américaines, à partir du milieu du 20<sup>ème</sup> siècle, Bogotá a subi des processus de croissance accélérée. Cette croissance urbaine accélérée est en grande partie due à un exode rural très fort motivé par des raisons économiques (creusement de l'écart en terme de dynamisme économique entre les villes et les campagnes) mais aussi par l'instabilité politique. La Colombie est en effet plongée depuis les années 1940 dans un conflit armé d'envergure nationale, qui affecte principalement les campagnes peu accessibles et avec une faible présence de l'Etat [Pécaut, 2003]. Dans ces conditions, un grand nombre de paysans se sont déplacés vers les capitales régionales et tout particulièrement vers Bogotá. Ce processus d'exode rural a été le moteur de la croissance, mais aussi de la ségrégation urbaine de la ville de Bogotá. Des processus complexes d'occupation de l'espace se sont développés, en lien notamment avec la capacité économique des divers groupes sociaux. Il existe à Bogotá une



segmentation forte de la production de l'espace [Parías, 2001]: les groupes sociaux aisés peuvent accéder au marché du logement par les voies légales, tandis que les groupes plus défavorisés sont obligés de recourir à des processus d'urbanisation de type « informel ». Ces processus informels ont pris une part importante dans la production de l'espace urbain. Ils se présentent sous des formes diverses : auto-construction, lotissement illégaux, invasion de terrains, etc. Ces processus ont tissé un paysage urbain complexe caractérisé par des formes de ségrégation présentes à l'échelle de la ville et à l'échelle micro des quartiers [Dureau et *al.*, 2000]. On observe ainsi à Bogotá, une complexification de l'organisation territoriale accompagnée d'un renforcement de la polarisation socio-spatiale [Dureau et *al.*, 2000]. Le lien existant entre les groupes sociaux et le processus de production de l'espace suggère que l'on doit accorder une place importante aux groupes sociaux dans la modélisation des dynamiques résidentielles à Bogotá.

Les travaux déjà réalisés sur Bogotá (Piron et *al.* 2004, 2006 ; Piron 2005) sur lesquels nous nous appuyons, ont démontré la pertinence de la considération des groupes sociaux mais aussi des groupes de logements (types d'habitat) comme entités principales de modélisation des dynamiques résidentielles de la ville de Bogotá. Toutefois, dans ces travaux, les groupes sociaux et les types d'habitat sont considérés de manière statique : leur nombre et leur caractéristiques ne changent pas au cours de la période d'étude. Nous proposons ici des mécanismes qui permettent de prendre en compte la formation automatique de ces groupes et leur évolution. Ainsi, nous prenons en compte les dynamiques de mobilité résidentielle intra-urbaine combinée à des processus démographiques globaux (croissance réelle et solde migratoire), comme moteur explicatif de la recomposition permanente du tissu urbain.

## **2. Enjeux dans la modélisation multi-agent.**

Ce travail s'inscrit dans le prolongement des travaux de (Mullon et al., 2001 ; Piron et al, 2003, 2007) qui ont développé une approche synthétique et globale pour modéliser les dynamiques des mobilités résidentielles intra-urbaines de Bogota en proposant la mise en place de niveaux d'abstraction intermédiaires de modélisation. Ces niveaux intègrent à la fois une approche agrégée pour rendre compte de l'évolution de la distribution de la population et des reconfigurations de la ville et une approche individuelle pour mettre l'accent sur les causes et l'impact des choix résidentiels sur la mobilité intra-urbaine. Le parti pris de la modélisation est donc de travailler sur les grandes structures de la composition sociale et de l'habitat de la ville, obtenues à partir de groupes localisés et pertinents de ménages et de logements. Cette modélisation présente néanmoins l'inconvénient de considérer des entités de modélisation statiques sur la période considérée.

Dans notre approche, contrairement aux approches classiques (où il s'agit de considérer l'individu ou le ménage), nous proposons de considérer et de faire évoluer des groupes de ménages (groupes sociaux) et des groupes de logements (types d'habitat) comme entités principales de modélisation. Nous considérons un niveau intermédiaire de modélisation qui a pour objectif de simplifier la définition et l'exécution d'interactions entre entités. En effet, lorsqu'on se situe à un niveau intermédiaire, les interactions entre entités répondent à des logiques globales moins nombreuses et plus faciles à identifier et à modéliser que les interactions au niveau microscopique. Les entités intermédiaires sont représentées dans le système proposé ici sous forme d'*agents composés abstraits* : *composés* car l'approche utilisée pour leur formation est proche de la formation de groupes ou coalitions d'agents. Ils sont composés d'un nombre important d'agents qui se situent à un niveau de modélisation inférieur; *abstrait* parce qu'ils représentent des abstractions de la réalité. Les agents composés agissent entre eux mais aussi avec les agents de(s) niveau(x) inférieur(s). Il existe donc des interactions à plusieurs niveaux mais aussi entre plusieurs niveaux.

Dans notre travail, nous nous intéressons tout particulièrement à la constitution automatique, à l'évolution et au comportement (en particulier, à la nature de leurs interactions) des agents

composés abstraits. Ces problèmes, du moins dans le cas d'agents composés abstraits, ont été très peu traités dans la littérature multi-agent. Plusieurs difficultés se présentent, notamment dans la définition des règles de constitution des ces agents, mais aussi dans la définition de leurs interactions. D'autre part, la constitution des agents composés fait intervenir un nombre très important de micro-agents.

## **2.1. Génération et manipulation automatique d'agents composés abstraits : Un problème multi-échelle.**

La modélisation d'un système social implique l'intervention de dynamiques à différentes échelles spatio-temporelles. Contrairement à la majorité d'exemples de simulation multi-agent, où seulement deux niveaux d'analyse sont considérés (le niveau microscopique, où les agents sont situés et simulés, et le niveau macroscopique qui est le niveau d'analyse des structures ou propriétés émergentes), dans la simulation social, il faut souvent considérer des acteurs à différents niveaux avec des points de vue et temporalités différents. Par exemple dans le cas urbain, les habitants, les aménageurs, les institutions ont des points de vue (échelles spatiales d'analyse ou d'action) différents de ce qui est la ville, et des phénomènes qui y agissent. Leur notion d'environnement et leurs actions ont des conséquences spatiales et temporelles à différentes échelles. En conséquence, quand il s'agit d'étudier ce type de systèmes, il faut considérer généralement plus de deux niveaux de modélisation, avec des agents hétérogènes qui interagissent entre eux à des échelles spatiales et temporelles différentes. Dans certains cas, des nouveaux objets plus « artificiels » sont introduits. C'est le cas lorsque l'on veut réduire la complexité par l'introduction d'abstractions comme des catégories d'individus, des groupes sociaux, des types d'habitat, etc. Ces objets opèrent normalement à des échelles qui se situent entre l'échelle micro et l'échelle macro. Les agents composés abstraits dont nous traitons dans ce document, sont des représentations de ce type d'abstractions.

D'un autre côté, le problème de la génération automatique et de la manipulation d'agents composés a été très peu étudié (Servat, 2000 ; Duhaut, 2002 ; Tranouez et *al.*, 2003 ; Bonté, 2005 ). De plus, les travaux sur ce sujet se centrent sur l'analyse des systèmes biologiques ou physiques bien définis. Dans ces cas, les agents composés qui sont les objets d'étude, sont des

représentations des structures physiques (des structures hydrologiques dans le cas de Servat) ou des organismes biologiques (des pseudo-organismes biologiques formés par des amibes<sup>1</sup> étudiés par Bonté). Ces agents composés répondent à des lois physiques ou biologiques plus ou moins bien connues. Ces lois dépendent, en général, du comportement des agents qui constituent l'agent composé. Au contraire, quand on veut modéliser des structures composées telles que les regroupements d'individus similaires<sup>2</sup>, des coalitions d'agents ou des réseaux d'influence, leurs comportements sont moins bien définis, car ce type de structures sont des abstractions sujettes à la subjectivité. Ces abstractions sont utilisées à des fins diverses, mais en général, elles offrent une réduction du phénomène étudié pour faciliter sa description ou son explication.

Nous avons identifié plusieurs problèmes qui sont liés à la génération automatique et à la manipulation d'agents composés (les abstraits et les moins abstraits) :

- Quelles interactions entre *agents composants*<sup>3</sup> donnent lieu à la création d'un agent composé? (problème de *formation*).
- Sous quelles conditions et/ou à quel moment un ensemble d'agents composants devient un agent composé? (problème d'*agentification*).
- Comment évoluent les agents composés?(problème d'*évolution*).
- Y-a-t-il des interactions entre agents composés ? Dans le cas affirmatif, quelles sont ces interactions ? (problème de *macro-interactions*).
- Comment les agents composés modifient/influent les comportements des agents qui les composent? (problème d'interactions *multi-niveaux*).
- Comment disparaissent les agents composés? (problème de *disparition*).

### **2.1.1. La constitution automatique d'agents composés**

La constitution d'agents composés comprend deux étapes : la formation et l'agentification. La formation des agents est une étape d'identification et d'agrégation d'un certain nombre

---

<sup>1</sup> Dans certains cas, des regroupements de *dictyostelium discoideum* (une sorte d'amibe unicellulaire) ont un comportement similaire à celui d'un organisme multicellulaire (cf. Bonté, 2005).

<sup>2</sup> Il faut noter que même s'il existe un grand nombre de mesures mathématiques de la ressemblance (cf. Nakache et Confais, 2003), le choix de la mesure à utiliser a une forte composante subjective, les regroupements seront d'autant plus abstraits que le choix de la mesure de ressemblance est subjective .

<sup>3</sup> Nous appelons agent composant, un agent qui intègre un ensemble d'agents qui composent un agent composé; l'agent composant est, en général, à un niveau inférieur de modélisation, par rapport à l'agent composé. Il faut noter qu'un agent composant peut être formé par des agents composants d'un niveau inférieur, on peut ainsi avoir une hiérarchie d'agents composés.

d'agents composants. L'agentification, dans ce cas, est le processus par lequel un agrégat d'agents composants se transforme en agent composé.

Le problème de la *formation* d'agents composés est lié au concept d'émergence dans le sens « pratique » du terme. En effet, une grande partie des travaux qui étudient l'émergence des phénomènes ou structures globales, s'intéressent à l'ajustement d'un ensemble d'interactions au niveau microscopique. Le but du jeu est de trouver les interactions (en général les plus simples) qui produisent tel ou tel phénomène/structure au niveau macroscopique. Il est donc difficile d'avoir une règle générale pour la détermination de ces interactions, car elles sont très dépendantes du problème étudié. La seule opinion en commun, que la majorité des auteurs semble partager, est la recherche des interactions les plus simples possibles (sorte de principe de « rasoir d'Occam » modifié, où on ne recherche pas l'explication la plus simple possible du phénomène, mais sa cause la plus simple). Ce qui se justifie, bien évidemment, d'une part par le besoin de simplifier des systèmes déjà « complexes », et d'autre part, par les résultats expérimentaux dans les différents domaines d'étude.

Le problème de l'agentification automatique, à la différence du problème de la formation ou de l'émergence de structures est très peu étudié. La plupart des études d'émergence de structures à un niveau macro, se limitent à faire une analyse *a posteriori*. Ces structures sont, en général, obtenues à partir de simulations de « sociétés » d'agents élémentaires qui interagissent entre eux grâce à des comportements prédéfinis très simples (par exemple : des études du vol d'oiseaux [Reynolds, 1987], des sociétés d'agents fourmi [Drogoul, 1993], des études de formation de molécules, etc.). La détection et l'analyse des structures émergentes (des chemins de fourmis, des molécules, des groupes d'oiseaux qui volent ensemble, etc.) est réalisé de façon « manuelle » et *a posteriori* par inspection visuelle de la distribution des agents simulés. Ce processus, visuel et personnel, est nécessairement influencé par la subjectivité. L. Bonté [Bonté, 2005] l'illustre dans son travail en donnant l'exemple suivant : « Combien de fois avez-vous regardé un nuage avec une autre personne, vous dites – Regarde, ce nuage a une forme de vache – et l'autre personne vous répond – Je dirais que c'est plutôt un chat ». Cet exemple laisse entrevoir, toutefois, une partie d'objectivité dans l'observation. Il existe en effet une forme, mais c'est à l'observateur de décider laquelle. Dans le cas des observations faites sur les simulations des systèmes complexes, il existe une combinaison entre la « réalité » (objectivité) de l'émergence d'une propriété macroscopique et la subjectivité de la description/analyse de cette propriété. Cette description a la tendance à être guidée par une intentionnalité donnée au système par son concepteur. Le concepteur veut modéliser tel ou tel phénomène, et son analyse des résultats est sûrement biaisée par son

objectif. Comme le constate A. Drogoul : « l'intentionnalité d'un système multi-agent n'est pas complètement définie par les agents, ni par l'observateur mais n'émerge que lorsque le système manifeste des propriétés que l'observateur reconnaît » ([Drogoul, 2000]). L'analyse (détection et explication) des structures émergentes est donc une tâche complexe et demande un effort très grand d'interprétation.

Dans les exemples disponibles qui tiennent compte de l'agentification automatique (Servat, 2000 ; Duhaut, 2002 ; Tranouez et *al.*, 2003 ; Bonté, 2005 ), les ensembles d'agents composants deviennent des agents composés lorsque un ensemble de conditions physiques ou biologiques sont vérifiées. Il s'agit par exemple dans le travail de D. Servat, de structures hydrologiques qui sont formées à partir des principes de la dynamique des fluides et qui tiennent compte de l'environnement sous la forme de la topographie du terrain. Ou encore dans le travail de L. Bonté, il s'agit de reproduire la formation d'un organisme pluri-cellulaire à partir des organismes plus simples. Dans ces exemples, l'agentification est gouvernée par des règles très bien définies qui sont liées au problème à résoudre. La situation est plus difficile quand il s'agit d'agentification d'agents composés abstraits. Les conditions d'agentification ne peuvent pas suivre des règles physiques, ni biologiques. Il faut avoir recours à des critères aussi abstraits que les structures. Ces critères quand ils sont quantifiables sont en général de type statistique (par exemple : critères de cohésion des réseaux, mesures d'agrégation d'individus semblables, etc.). Le choix du critère n'est pas très facile, car très souvent, un grand nombre des critères différents est possible.

### **2.1.2. Les interactions à différents niveaux et l'évolution des agents composés.**

Les interactions entre agents dépendent de leur degré de proactivité (voir caractéristiques des agents, section 2.2.2.4) et de leur structure/définition interne. D'une part, plus un agent est proactif, plus il aura la tendance d'interagir et de modifier son environnement de manière autonome. D'autre part, les réponses aux interactions et à l'état de l'environnement dépendent de la complexité des mécanismes internes de l'agent. Ainsi, par exemple : des agents tels que les agents réactifs [Drogoul & Ferber, 1994] traitent les stimulus externes par des mécanismes internes basés sur des règles très simples d'action/réaction (on garantit toujours qu'un même stimulus produit toujours une même réponse). Au contraire, des agents cognitifs [Maes, 1990], ont des systèmes très complexes qui combinent des caractéristiques « humaines » tels que croyances, désirs, connaissances, capacité d'apprentissage, etc. Un agent de ce type ne

répond pas forcément de la même façon au même stimulus, car les croyances et/ou connaissances que l'agent a de son environnement sont en constante évolution.

L'évolution des agents dépend de leur complexité. Dans le cas des agents simples ou minimalistes, seules les valeurs des états ou propriétés internes peuvent changer : leur comportement individuel ne change pas (dans l'approche minimaliste on s'intéresse plus à l'évolution du système qu'à l'évolution individuelle de chaque agent). Dans le cas des agents complexes, en plus des changements dans les états internes, le comportement individuel des agents peut évoluer par des systèmes d'adaptation et apprentissage de l'environnement.

Dans ce travail, nous nous intéressons à des agents minimalistes qui peuvent former des agents composés relativement complexes. Dans ce cadre, il faut s'interroger sur la capacité d'intervention (ainsi que la pertinence de cette intervention) du modélisateur dans la définition des mécanismes d'interaction et d'évolution des agents générés automatiquement. Il faut trouver un compromis entre ce qui est prévu et ce qui émerge du système.

## **2.2. Un grand nombre d'agents : simulations à large échelle.**

Quand on s'intéresse au comportement de la distribution de la population d'une ville, on se trouve vite confronté au problème du nombre important de composants qui peuvent intervenir (habitants, logements, infrastructure, structures politiques, organisations, etc.). Ce problème qui en principe est d'ordre technique, détermine non seulement la taille des ressources informatiques nécessaires à la simulation, mais aussi la faisabilité de la simulation en termes de temps de calcul, ainsi que la faisabilité de l'interprétation des résultats (celle-ci dépend de la complexité des résultats obtenus). Ce problème devient alors un aspect fondamental de la modélisation et peut même conditionner le choix de la problématique.

Plusieurs mesures « techniques », cherchant à réduire la quantité d'information utilisée, peuvent être prises. La devise dans ces procédures est : « utiliser autant de détail qu'il est possible, mais seulement considérer le détail nécessaire à produire un certain comportement émergent » [Schweitzer, 2003]. Certaines procédures statistiques peuvent être utilisées pour pré-traiter les données qui décrivent le système d'étude. Parmi ces procédures, il est possible de faire par exemple, une analyse de la pertinence des variables utilisées pour réduire leur nombre au strictement nécessaire, ou encore un échantillonnage sur les données de manière à

avoir un nombre d'agents plus réduit que le nombre total. Mais cet ensemble réduit doit toujours être représentatif de l'ensemble total et l'on doit être capable de reproduire les dynamiques que l'on veut étudier.

Au niveau de la modélisation, d'autres mesures cherchant à réduire la complexité des dynamiques peuvent être prises. Il est possible d'essayer de trouver l'ensemble des règles d'interaction (entre agents individuels) les plus simples et d'utiliser des abstractions à différents niveaux de modélisation. Pour trouver l'ensemble des règles les plus simples possibles, on peut utiliser une approche incrémentale (comme celle proposée par [Schweitzer, 2003] dans le cadre de la conception d'environnements d'agents minimalistes). Il s'agit de commencer par des modèles très simples et très proches de modèles physiques élémentaires pour tester pas à pas des interactions plus complexes au niveau microscopique en se rapprochant de plus en plus du phénomène macroscopique désiré.

L'utilisation d'abstractions à différents niveaux de modélisation peut se faire par réduction de la granularité de façon globale ou locale. En fonction du problème étudié, on peut réduire le niveau de granularité en représentant certaines parties ou sous-systèmes par un seul agent, au lieu d'avoir des agents qui représentent des composantes plus élémentaires. Parfois, on peut réduire la granularité générale du système étudié. Par exemple, dans le cas de l'analyse de la dynamique des mobilités résidentielles dans une ville, il est souvent judicieux de représenter non pas les individus, mais les ménages. Parfois, il est judicieux de réduire la granularité de certaines parties du système seulement, en gardant un niveau de granularité plus détaillé dans d'autres parties.

L'approche proposée ici, basée sur la création d'agents composés est une manière de réduire la granularité du système de façon dynamique, tout en gardant la possibilité de revenir à des niveaux de granularité plus fins.

Dans la partie I de ce document, nous présentons une revue de l'état de l'art en modélisation des dynamiques urbaines et en formation de structures composées (abstraites et moins abstraites). Dans la partie II, nous présentons deux propositions de modèles pour reproduire la distribution de la population spatiale d'une ville. Ces modèles ont fait l'objet de plusieurs publications (Gil-Quijano et *al.*, 2007a ; 2007b, 2007c). Le premier de ces modèles est dit synthétique et ne tient compte que des groupes sociaux et des types d'habitat. Le deuxième prend en compte les groupes sociaux et les types d'habitat, mais aussi le niveau microscopique formé par les ménages et logements. Dans la partie III, nous présentons les



résultats de simulations réalisées avec les deux modèles sur des données de la ville de Bogota. Dans cette partie-là, nous procédons à une analyse de la validité de chacun de ces deux modèles. Finalement, dans la partie IV, nous discutons la problématique présentée dans cette introduction, et évaluons la pertinence des modèles proposés et des résultats obtenus dans les simulations.

# **Partie I: Etat de l'art**

# **Chapitre I : Modélisation urbaine et simulation**

## **1.1. Introduction aux notions de modélisation et de simulation**

Un des objectifs principaux de la démarche scientifique est de comprendre la réalité. Pour cela, on identifie des phénomènes que l'on essaie d'expliquer/comprendre par différents moyens. Un phénomène (sans rentrer dans le débat philosophique qui découle de sa définition) est le résultat d'une observation de la réalité, réalisée par un sujet. Selon la définition de Kant qui oppose le phénomène au « noumène » (la réalité en soi qui est indéfinissable et indépendante de notre expérience), les phénomènes constituent la réalité telle que nous la percevons. Une de manières d'étudier des phénomènes est de construire un modèle, c'est-à-dire une représentation simplifiée des phénomènes observés qui doit être « pertinente » et « maniable » [Coquillard & Hill, 1997]:

- pertinente, car le modèle doit se comporter, avec une exactitude acceptable, comme les phénomènes représentés se comporteraient,
- maniable, car la description donnée par le modèle doit être plus appréhendable que les phénomènes qu'il représente,

Pour construire cette simplification maniable et pertinente, on choisit certaines données représentatives et on fait un certain nombre d'hypothèses sur les comportements des phénomènes étudiés. A partir des données sélectionnées, on définit des variables et/ou des éléments qui décrivent ou participent des phénomènes étudiés. Les valeurs des caractéristiques instantanées des différents éléments ou les valeurs instantanées des variables considérées constituent « l'état » du modèle.

La nature des modèles varie selon leur finalité. Ils peuvent être théoriques (ex. : pour aider à comprendre une organisation ou des dynamiques socio-spatiales), opérationnels (ex. : pour la gestion des territoires et à la prise de décision), ou bien prédictifs (ex. : pour essayer de prévoir les développements futurs des villes).

Pour étudier l'évolution des phénomènes dans le temps, on effectue des « simulations » sur le modèle qui représente les phénomènes. Une simulation consiste à retracer une partie de l'évolution dans le temps d'un modèle, c'est-à-dire, à retracer les changements des états du

modèle sous certaines conditions qui découlent des hypothèses faites sur le phénomène. L'évolution du modèle peut être calculée de deux manières différentes. La première consiste à considérer le temps de manière discrète, c'est-à-dire en le découpant en intervalles de la même longueur (pas de temps) et en actualisant l'état du modèle de manière successive à chaque pas de temps. La deuxième manière est de considérer des événements discrets et de faire changer l'état du modèle par l'exécution successive de ces événements.

Les simulations ont deux types d'objectif :

- elles peuvent servir à la recherche de l'explication d'un phénomène : les simulations ne permettent pas de trouver directement une explication des causes d'un phénomène, mais permettent de tester plusieurs explications afin de trouver celle qui permet de mieux reproduire le phénomène ;
- elles servent d'outil de prédiction à partir des résultats obtenus à partir de la définition de scénarii sur les caractéristiques du phénomène étudié.

Ainsi, les simulations permettent entre autre de tester différentes hypothèses sur le comportement d'un phénomène, sans « expérimenter » directement sur le phénomène lui-même. Cette caractéristique est particulièrement utile, lorsque les modifications directes sur le phénomène sont coûteuses (par exemple des expérimentations sur le crash d'un avion) difficiles, dangereuses, ou lorsque des problèmes éthiques se posent (notamment lorsqu'il s'agit d'expérimentations avec des groupes humains).

La simulation offre un avantage évident dans l'étude des phénomènes sociaux : elle permet de faire des « expérimentations » peu coûteuses et sans conséquences sur les populations étudiées sans les « toucher » directement. La simulation donne ainsi aux sciences sociales une capacité expérimentale qui n'était propre auparavant qu'aux sciences physiques, chimiques et dans une certaine mesure aux sciences biologiques. Grâce à la simulation et notamment à la simulation sur ordinateur, on peut construire des « mondes virtuels » qui sont gouvernés par des lois construites par le modélisateur à partir des observations du monde réel combinées aux hypothèses faites sur l'évolution des phénomènes étudiées. On peut ainsi représenter des phénomènes plus ou moins complexes, changer facilement les hypothèses, tester différents scénarii, changer le degré de finesse des données utilisées (utiliser plus ou moins de données, plus ou moins précises, pour s'approcher des phénomènes ou au contraire simplifier leur représentation) et observer les conséquences des ces changements. Ainsi, on peut par exemple prévoir les effets sociaux et/ou économiques sur une population donnée de la modification d'une politique d'imposition (exemple classique des méthodes de micro-simulation, cf. section 2.2.1.) ou étudier les conséquences des changements de l'infrastructure

d'un secteur d'une ville sur les dynamiques de mobilité à différents niveaux (par exemple les effets de la construction d'une nouvelle route sur la circulation globale de la ville, ou les effets de la mise en place d'un feu rouge à un certain croisement sur la circulation locale d'un quartier).

Dans le cas de l'étude d'un phénomène par simulations, il faut d'abord construire un modèle, c'est-à-dire, déterminer les composantes, variables ou éléments pertinents ainsi que leur relation pour représenter le phénomène. Ensuite, il faut déterminer lesquels évoluent dans le temps et selon quelles règles ou fonctions. Dans le cas des phénomènes urbains, il est souvent judicieux de décomposer la réalité en différents niveaux de représentation plus ou moins détaillés, en la décomposant en composantes et éléments. Ce type de représentation s'adapte bien aux phénomènes urbains, car il faut souvent intervenir un nombre important d'acteurs hétérogènes (politiciens, aménageurs, citoyens, entreprises, etc.) qui ont des perceptions et des capacités d'action différents sur le phénomène représenté et qui habitent, évoluent et agissent dans un espace complexe, dont la complexité varie selon le niveau de représentation considérée (la ville, l'arrondissement, le secteur urbain, le quartier, le logement, etc.).

Dans notre travail, nous proposons un modèle théorique qui a pour objectif de reproduire la distribution spatiale de la population dans la ville de Bogotá. Grâce à des simulations, nous pouvons retracer cette distribution sur une période de temps donnée. Dans ce modèle, nous considérons comme entités principales de modélisation des groupes homogènes de ménages (groupes sociaux) et des groupes homogènes de logements (types d'habitat). Dans le cadre de l'écologie urbaine (Vicari, 1981 ; Grafmeyer & Joseph, 1990 ; Roncayolo 1990 ; Grafmeyer 1994 ), ces groupes sociaux sont en concurrence pour l'acquisition de l'espace urbain qui est représenté par les logements libres de certains types d'habitat. Les groupes sociaux et les types d'habitat offrent une description simplifiée de la réalité dont la pertinence a été démontrée dans les travaux précédents sur les dynamiques de mobilités résidentielles à Bogotá (Piron et al. 2004, 2006 ; Piron 2005). Les groupes sociaux et les types d'habitat sont décrits par un nombre important de variables, qui font que leur étude et leur représentation se situent dans un cadre proche de l'écologie factorielle urbaine (Berry et Kasarda, 1977; Abu Lughod, 1969 ; Préteceille et de Queiroz, 1999 ; Pruvot et Weber-Klein, 1984).

Dans le chapitre suivant, nous présentons une série de différents types de modèles urbains qui ont été développés jusqu'alors. Ils sont présentés de manière plus ou moins chronologique de manière à présenter leur évolution vers les modèles les plus récents pour terminer par une proposition de modèle adapté à la problématique étudiée dans ce document.

## 1.2. Modélisation et simulation en géographie

En géographie, même si dès 1954 Alain Reynaud publiait dans *l'Information géographique* un article intitulé « Modèles et idéal-type en géographie : l'exemple des fermes beauceronnes », les modèles ont commencé à être employés grâce au développement de l'analyse spatiale dans les années 1970, à la suite de l'ouvrage fondateur de Peter Haggett, *Locational Analysis in Human Geography* (1965). Il manifeste la volonté de faire accéder la géographie à des lois générales, ou tout du moins à un certain niveau de généralité : l'espace n'est plus seulement un support des phénomènes mais un agent physique dont les éléments, distance, gradients, rythmes jouent un rôle important. Cette « nouvelle géographie » prône une démarche déductive et la recherche de similarités et d'ordre dans les processus étudiés. Elle s'appuie sur l'étude de mécanismes généraux et sur la proposition de modèles. Il s'agit d'un changement de paradigme par rapport à la géographie classique idiographique qui vise l'étude des spécificités régionales, en partant du postulat de l'unicité des régions, et qui pratique une démarche inductive. L'« Ecole française de géographie » s'était ainsi constituée, dans la continuité des travaux de Paul Vidal de La Blache dans la première moitié du XXème siècle. Les principaux auteurs qui ont développé une approche modélisatrice en géographie sont Yves Guermond<sup>4</sup>, Denise Pumain<sup>5</sup>, Franck Auriac<sup>6</sup>, Philippe et Geneviève Pinchemel<sup>7</sup>, Roger Brunet<sup>8</sup>.

La simulation a commencé à être utilisée en géographie avec l'intégration explicite de l'espace aux modèles de simulation, notamment les modèles de microsimulation spatiale, par exemple le modèle SVERIGE (Holm et al., 2002 ; Holm & Sanders., 2007 cf. section 2.2.2.2), les automates cellulaires (Schelling, 1971 ; Almeida et al., 2002 ; Batty et al. 1999 ; Coquillard & Hill, 1997 ; Kirtland et al, 1994 ; Benenson 1998; Schofisch & Haderler 1996; Wahle et al. 2001 ; cf. section 2.2.2.3) et la simulation avec des systèmes multi-agents (Benenson 1998; Portugali et al. 1997; Sanders et al. 1997 ; Daudé, 2002 ; cf. section 2.2.2.4). Nous présentons ces différents types de modèles d'une manière plus approfondie dans le chapitre suivant.

---

<sup>4</sup> GUERMOND Y. (1984), *Analyse des systèmes en géographie*, collection Sciences des systèmes, Presses Universitaires de Lyon.

<sup>5</sup> PUMAIN D. (1996), *Penser la ville. Théories et modèles*, Anthropos - thème

<sup>6</sup> AURIAC F., *Système économique et espace* (1979), Thèse d'Etat sur la schématisation systémique de l'évolution du vignoble languedocien.

<sup>7</sup> PINCHEMEL P. et PINCHEMEL G. (1988), *La face de la Terre. Eléments de géographie*, Paris : Armand Colin.

<sup>8</sup> BRUNET R. (1990), « la dynamique des systèmes » chapitre IX du livre premier *Le déchiffrement du monde* de BRUNET R. & DOLLFUSS O., Mondes nouveaux, Geo Universelle, vol. 1, Belin-reclus.

## **Chapitre II : Modélisation informatique des dynamiques urbaines**

Dans ce chapitre, nous faisons une revue rapide des modèles de simulation utilisés dans l'étude des dynamiques urbaines. Nous considérons deux types de modèles. Les premiers considèrent un seul niveau de modélisation ; les seconds proposent une approche à plusieurs niveaux (au moins et généralement deux niveaux) de modélisation. Nous nous intéressons particulièrement à cette deuxième famille de modèles et en faisons une présentation plus approfondie. À la fin du chapitre, nous concluons sur les avantages et les inconvénients des différentes approches et faisons une proposition de modélisation pour notre cas d'étude.

### **2.1. Modélisation sur un niveau**

Les premières formes de modélisation de l'évolution d'un système qui ont été mises au point dans les années 1950 utilisent des *méthodes numériques*. Dans cette approche, on ne considère que des systèmes dont l'évolution est déterministe et linéaire (causalité linéaire). Le système est décrit à l'aide d'un nombre important d'attributs. Ces attributs sont de deux types: des variables d'état (qui représentent l'état du système) et des taux de changement (pour représenter les changements entre les états). La dynamique du système est décrite à l'aide d'ensembles d'équations différentielles qui tiennent compte des variables d'état et des taux de changement. Le temps est considéré de façon discrète, ce qui permet de trouver une solution numérique au système d'équations différentielles (dans les systèmes d'équations différentielles classiques, le temps est considéré comme continu). Les méthodes numériques donnent une vision macroscopique des dynamiques du système: on considère que la dynamique du système peut être exprimée simplement à partir des relations entre les variables globales. Les entités qui composent le système ne sont pas considérées. Cette vision présente plusieurs inconvénients [Drogoul et Ferber, 1992]. Le principal est qu'il n'est pas possible de

modéliser des mécanismes de rétroaction parce qu'il est impossible de représenter les actions individuelles ainsi que leurs relations avec les variables globales.

L'approche de Forrester [Forrester, 1968] est une des premières propositions pour tenir compte des multiples interactions et rétroactions entre les différentes composantes d'un système, qui sont la source d'effets « contre-intuitifs » dans les systèmes sociaux. Forrester considère que ce caractère contre-intuitif est une des raisons de l'échec des politiques urbaines: les politiques urbaines ne peuvent que produire des effets inattendus car elles sont pensées dans une approche cause-effet, avec une vision temporelle limitée et ne tiennent pas compte des interactions entre les multiples variables socio-économiques [Forrester, 1971]. Ainsi, Forrester propose un formalisme inspiré de l'hydraulique pour modéliser la dynamique des systèmes sociaux. Des variables globales sont représentées sous forme de réservoirs. Le contenu des réservoirs peut représenter des individus, des informations ou tout objet qui entre et/ou sort du système. La quantité contenue dans les réservoirs varie dans le temps en fonction des flux entrants et sortants. Ces flux sont contrôlés par des vannes qui représentent des centres de décision, qui reçoivent des informations et les transforment en actions. Les flux transitent par un canal d'information qui relie les variables d'état et permet de simuler des rétroactions. Le comportement des réservoirs est représenté mathématiquement par des équations différentielles qui relient les flux entrants et sortants de chaque stock. De cette manière, le contenu des réservoirs représente l'état instantané du système, qui est modifié par l'intervention des flux et les interactions entre les différents composants du système. L'approche forrestienne associe à chaque modèle un diagramme sagittal (voir exemple dans la figure ci-dessous) qui montre les relations entre toutes les variables de manière graphique.

Dans ce cadre, Forrester a développé le modèle de « dynamique urbaine » (Urban Dynamics [Forrester, 1969]) qui a un double objectif: reproduire le cycle de vie des aires urbaines sur une longue période de temps, explorer les conséquences lointaines des politiques urbaines alternatives. Le modèle considère trois grands sous-systèmes: les entreprises, les logements et la population active. Chaque sous-système est divisé en plusieurs catégories. Les trois sous-systèmes sont liés de manière très rigide par des correspondances étroites et fixes: les matrices d'emploi entre les catégories de la population active et les types d'entreprise ; les relations étroites entre les catégories de la population active et les types de logements (ex : les logements de haut-standing sont moins densifiés que les logements ordinaires, eux-mêmes moins peuplés que les taudis où s'entassent les travailleurs sans qualification et les chômeurs) ; les rapports entre la taille moyenne des familles et le degré de paupérisation. Les transformations du système sont apportées par des impulsions venues de l'extérieur : afflux de



travailleurs en provenance d'autres villes attirés par le niveau élevé des salaires, politique des logements à bon marché, pression fiscale modérée, création d'entreprises ou de logements suscités par de faibles valeurs foncières ou une politique urbaine incitative, etc. Le système s'auto-régule grâce aux flux internes des variables globales jusqu'à ce qu'il atteigne un niveau d'équilibre stable.

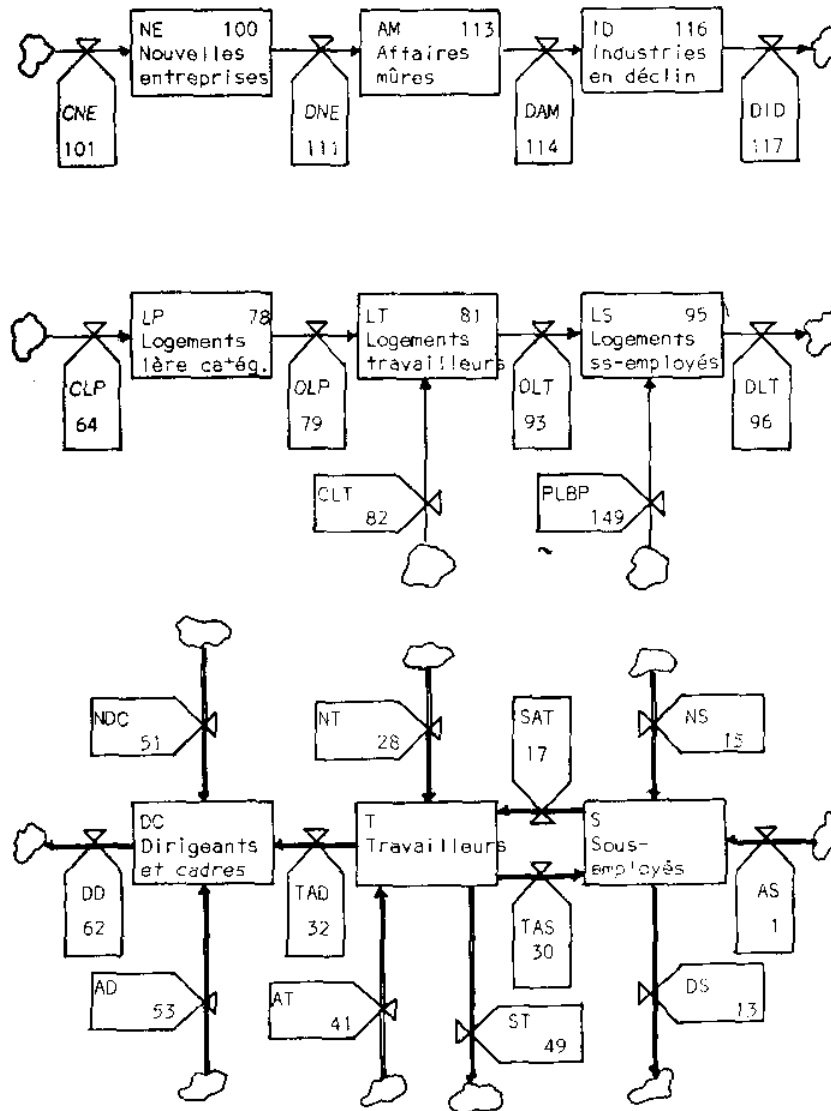


Figure 1. Exemple de diagramme sagittal dans une modélisation par l'approche forrestrienne. Le diagramme présente les trois sous-systèmes considérés dans le modèle urbain de Forrester, les rectangles représentent les 9 principales variables globales considérées, les vannes représentent les 22 principaux taux de changements. Source [Forrester, 1979]

Le modèle urbain de Forrester a parfois été accusé d'être trop simpliste en raison de sa vision « caricaturale » des interactions sociales. On lui a reproché parfois de montrer une « prise de position trop extrême » avec des relations causales mal identifiées et des paramètres arbitraires sans correspondance avec la réalité (Derycke & Franck, 1977; Paelinck, 1972 ; Rothenberg, 1974 ; Stern & Bursaux, 1971 ; Alègre et *al.*, 1975, Prud'Homme et *al.*, 1972). Nous considérons que la principale limite de l'approche forrestienne est qu'elle modélise le système à partir de plusieurs sous-ensembles indivisibles: on ne peut pas considérer des individus ni différents niveaux de modélisation. D'autre part, le modèle de dynamique urbaine proposé par Forrester est très complexe en raison de l'énorme nombre de boucles d'interactions considérées.

## **2.2. Modélisation avec plusieurs niveaux**

Quand il s'agit de l'étude de systèmes sociaux, les systèmes de type forrestiens s'avèrent limités à cause de leur approche mono-niveau. En effet, lorsque l'on s'intéresse à des systèmes sociaux, il est souvent important de considérer différents niveaux de modélisation, avec des composantes (individus, ménages, voitures, mais aussi des groupes, des organisations, entreprises, etc.) qui agissent à différents niveaux, pas seulement le niveau macroscopique, et qui influent sur le comportement global du système. Dans cette partie, nous présentons quelques approches qui s'attaquent au problème de la modélisation multi-niveau et qui tiennent compte des différents composants présents dans les systèmes sociaux.

### **2.2.1. Approches descendantes : La microsimulation**

La microsimulation propose une première approche pour tenir compte des relations entre le comportement global d'un système social et le comportement des entités qui le composent. L'approche proposée dans la microsimulation est dite descendante : étant donné un comportement global, on souhaite comprendre les effets dans les comportements de la population d'entités du système. Les modèles de microsimulation sont normalement utilisés

pour tester les effets dans le temps des changements dans des politiques économiques sur une population donnée.

Le précurseur des idées qui sont à la base de la microsimulation (ou approche micro-analytique) est l'économiste Guy Orcutt (Orcutt, 1957 ; Orcutt et *al.*, 1961). Orcutt et ses collaborateurs ont développé plusieurs modèles qui avaient pour objectif de modéliser différents aspects de l'économie états-unienne. Ces modèles proposent une approche individuelle qui s'oppose à l'approche macro-économique traditionnelle de l'époque (Tinbergen, 1939 ; Stone, 1966 ; Leontief, 1951). Le besoin d'inclure les individus naît notamment de l'incapacité des modèles macro-économiques à évaluer les effets des changements des politiques économiques ou sociales dans les unités qui prennent des décisions, telles que les individus, les sociétés ou les ménages.

Dans la microsimulation classique, on considère au moins deux niveaux de modélisation : le niveau microscopique composé par des micro-entités telles qu'individus ou ménages, et un niveau agrégé (par exemple la population d'une région ou d'un pays). Le temps est discrétisé et, en général, chaque pas de temps représente une année. De cette manière, dans les modèles classiques de microsimulation, l'accent est mis dans l'identification et la représentation des preneurs de décision individuels (les micro-entités) et de la modélisation d'un comportement dynamique et adaptatif capable de reproduire les réponses d'un nombre important d'individus à des influences externes, notamment les changements des variables globales [Krupp, 1986].

Les micro-entités sont décrites par un certain nombre d'attributs (par exemple si les micro-entités sont des individus, ces attributs peuvent être l'âge, le genre, le niveau d'éducation, le statut de travail, etc.). Chaque micro-entité a un état défini par les différentes valeurs des attributs de l'entité. À chaque pas de temps, l'état de chaque entité est actualisé selon des règles de transition probabilistes. La microsimulation offre donc une approche stochastique pour décrire l'évolution d'un système qui s'oppose à l'approche déterministe des systèmes forrestiens. Dans une microsimulation, tous les individus, ménages ou entreprises qui composent la population réelle, ne sont pas nécessairement représentés. Normalement, on n'effectue des simulations que sur un échantillon représentatif de la population. Ainsi, à la fin de chaque simulation une projection est faite à partir des résultats obtenus sur l'échantillon initial, afin de pouvoir les appliquer sur la population totale.

Un des modèles les plus représentatifs de la microsimulation classique est le modèle DYNASIM (Dynamic Simulation Of Income Model) développé par Orcutt [Orcutt et *al.* 1976]. L'objectif de DYNASIM est de simuler le comportement économique et social des ménages états-uniens au gré des changements des variables globales qui représentent le

modèle macro-économique des Etats-Unis (taux global de chômage, salaires moyens, etc.). Les caractéristiques qui définissent les ménages se basent sur le profil socio-économique du chef de ménage (âge, groupe racial, genre, secteur de travail, niveau d'éducation, etc.). Dans ce modèle, on considère 15 événements possibles définis par des règles de transition individuelles (par exemple : mariage, naissance d'un nouveau membre du ménage, changement de travail, etc.). Le déclenchement d'un événement dépend d'une probabilité de changement d'état. Cette probabilité dépend à la fois de l'état courant du ménage et des valeurs des variables économiques globales.

Les recherches sur le modèle DYNASIM ont été à la base du développement d'un autre modèle de microsimulation : le modèle CORSIM, développé par une équipe de recherche de la Cornell University sous la direction de Steven Caldwell (Caldwell & Keister, 1996 ; Shaw, 2000). Ces travaux ont mené à l'inclusion de 25 événements et de plus de 700 règles de transition. CORSIM est conçu pour analyser des scénari de réforme de la sécurité sociale, d'accès aux soins dentaires, de politiques de retraites.

Les modèles de microsimulation classiques sont très gourmands quant à la quantité de données nécessaires à la simulation. La détermination des règles de transition requiert des analyses statistiques additionnelles sur la population d'étude. Ces règles sont difficilement généralisables à d'autres populations. D'autre part, il est très difficile de simuler les effets de rétroaction sur les variables globales : dans la microsimulation classique, on définit de façon statique la valeur d'une variable globale et l'on étudie les effets de cette variable sur la population. Les effets des dynamiques de la population sur la variable globale ne sont pas pris en compte. Par exemple, avec la microsimulation, on peut étudier la réponse des micro-entités (qui représentent une population) à un changement dans la politique d'imposition ; mais les politiques macroscopiques n'évoluent pas en fonction des réactions des micro-entités. Elles ne peuvent changer que par l'intervention du modélisateur. Une dernière limite des modèles de microsimulation est qu'ils sont déconnectés de leur environnement [Boman et Holm, 2005], c'est-à-dire qu'il n'existe aucune définition de l'espace géographique et aucune interaction n'est modélisée entre les entités. Pour cette dernière raison, les modèles de microsimulation classique ont été très peu utilisés dans la modélisation de problèmes où la composante spatiale est très importante [Clarke, 1996].

### **2.2.2. Approches ascendantes : Les systèmes auto-organisés**

Dans la section précédente, nous avons constaté que les principales limites des modèles de microsimulation classique sont :

- l'impossibilité de permettre aux individus d'influer sur l'évolution des variables globales (pas de boucles de rétroaction);
- l'absence de représentation de l'espace ;
- l'absence d'interaction entre les micro-entités

Dans cette partie, nous présentons plusieurs types de modèles qui proposent des solutions, parfois complémentaires, aux limites de la microsimulation classique. Ces modèles privilégient une approche dite ascendante qui permet de mettre en évidence les effets au niveau global des comportements des micro-entités individuelles. On met l'accent sur les interactions entre micro-entités comme génératrices de comportements globaux « inattendus » dits comportements émergents : ce sont les systèmes appelés « auto-organisés ». Dans ce type de systèmes, il existe une représentation explicite de l'espace. Les entités « habitent » dans cet espace et interagissent localement : les entités ne peuvent interagir qu'avec un petit nombre d'autres entités localisées dans un voisinage restreint.

#### 2.2.2.1. Modèles évolués de microsimulation

Plusieurs modèles se sont développés autour du problème de la représentation de l'espace et de la modélisation des mécanismes de rétroaction.

Un de ces modèles est MIMOSE [Möhring, 1996]. Dans MIMOSE, les micro-entités sont des objets qui interagissent avec un environnement. L'environnement est un objet qui se situe au niveau agrégé : il est défini par les variables agrégées qui décrivent le système. La grande avancée par rapport à la microsimulation classique réside dans le fait que les micro-entités peuvent modifier les valeurs des variables agrégées, via la modification de leur environnement : à chaque pas de temps, chaque micro-entité évalue son environnement et réagit en le modifiant selon un comportement prédéfini. Un mécanisme de rétroaction est ainsi implémenté. La représentation sous forme d'objets permet de tenir compte des événements de naissance et de mort des micro-entités. L'approche proposée dans MIMOSE s'adapte très bien à des problèmes où il est suffisant de considérer des interactions entre les micro-entités et le niveau agrégé. Lorsque l'on s'intéresse à des problèmes où les interactions locales entre micro-entités doivent être modélisées, cette approche s'avère insuffisante.

### 2.2.2.2. La microsimulation spatiale

La microsimulation spatiale est une technique de modélisation qui naît du croisement entre la microsimulation classique et les travaux de la *time geography* [Hägerstrand, 1975a ;1975b ;1995]. Dans la microsimulation spatiale, on s'intéresse à la représentation des acteurs, des ressources et d'autres objets localisés au niveau microscopique dont les interactions sont soumises à des contraintes spatio-temporelles. L'accent est mis sur l'importance de la localisation et de la durée des événements ou actions individuels. Les principales idées de la *time geography* adaptées par la microsimulation spatiale sont [prises de Boman et Holm 2005] :

- les actions sont issues du comportement d'acteurs individuels (personnes, ménages, entreprises) ;
- les acteurs, ressources et contraintes sont localisés dans l'espace, et agissent sur la succession des événements ;
- les actions et les événements sont influencés par les caractéristiques individuelles des acteurs, par le contexte environnemental et par les actions des autres acteurs ;
- la succession des événements est influencée par un facteur aléatoire ;
- il existe des interactions entre les acteurs individuels qui ont des intentions, des devoirs, qui élaborent des projets, et qui subissent des contraintes, dans un contexte spatio-temporel ;

Un des exemples de microsimulation spatiale est le modèle SVERIGE [Holm et *al.* 2002]. Ce modèle a été développé au *Spatial Modelling Center* à Kiruna, en Suède. Il est en partie inspiré du modèle CORSIM, mais à la différence de ce dernier, l'espace géographique est considéré de manière explicite. L'idée du projet SVERIGE est de créer un laboratoire artificiel d'expérimentation pour étudier l'effet des politiques gouvernementales ou municipales dans « un espace et un temps réels » [Bolman et Holm, 2005]. À cette fin, différents types d'entités ont été distingués : des individus, des firmes, des écoles, des agents de location de logements et des municipalités. Chaque pas de simulation correspond à un an. Un grand nombre d'événements sont considérés. Par exemple, les individus naissent, vont à l'école primaire, secondaire, à l'université, quittent leur famille, trouvent un travail, ont un salaire, changent de salaire, trouvent un partenaire dans un « marché de partenaires », se marient, divorcent, migrent dans des secteurs localisés et meurent. Les micro-entités peuvent créer ou détruire des liens selon les observations des autres micro-entités, par exemple les individus se marient ou divorcent. Quelques exemples des applications de SVERIGE sont présentées dans [Holm & Sanders, 2007] et [Holm et *al.*, 2002].

### 2.2.2.3. Les automates cellulaires

Les automates cellulaires proposent une technique de modélisation, où l'espace est la variable principale. Dans cette approche, le temps et l'espace sont discrets. L'espace est divisé en unités spatiales (cellules). Chaque cellule possède un ensemble de caractéristiques qui décrivent son état. L'évolution du système est le résultat de l'évolution individuelle et parallèle des cellules et dépend des règles générales appliquées au niveau *local*.

Un automate est une entité discrète par rapport au temps. Un automate est défini par un ensemble d'états internes qui évoluent selon un ensemble de règles faisant intervenir l'état courant de l'automate plus un certain nombre de signaux d'entrée provenant de l'extérieur de l'automate. Formellement, un automate peut être représenté par un ensemble d'états  $E$  et un ensemble de règles de transition  $R$ . Les règles de transition définissent l'état de l'automate  $E_{t+1}$  au temps  $t+1$  à partir de l'état  $E_t$  au temps  $t$  et l'ensemble de signaux d'entrée  $S$ .

Un automate cellulaire est un ensemble interconnecté d'automates individuels (cellules) localisés dans un espace partitionné. Les signaux d'entrée  $S$  nécessaires pour exécuter les règles de transition  $R$  de chaque cellule dépendent de l'état interne des cellules dans le voisinage. Les règles de transition sont les mêmes pour toutes les cellules. De cette manière, un automate cellulaire modélise un « monde » où l'espace et le temps sont discrétisés. Dans les applications de modélisation urbaine, la plupart du temps, les cellules représentent des unités spatiales telles qu'îlots, secteurs urbains, etc. Les automates cellulaires classiques sont formés par une grille uniforme (avec une, deux ou trois dimensions) de cellules identiques de forme rectangulaire ou hexagonale. Toutefois, les cellules peuvent être irrégulières et avoir des formes complexes telles que des partitions de Voronoi<sup>9</sup>, ou représenter des partitions obtenues à partir d'un système d'information géographique (SIG) (Benenson et al. 2002; O'Sullivan 2001; Sembloni 2000; Shi & Pang 2000 ; Batty et al. 1999 ; Kirtand et al, 1994). Parfois, même la forme de ces zones peut évoluer [Colonna et al. 1998].

Dans le cas des études urbaines pratiques, les automates cellulaires ont été utilisés principalement pour reproduire des problèmes de ségrégation [Schelling, 1971], reproduire la croissance urbaine (Almeida et al., 2002 ;Batty & Xie, 1994; Clarke et al., 1997; Xie & Batty, 2005 ;Besussi et al., 1998), l'évolution de l'usage du sol (White & Engelen, 1993; 1997) et le développement des centres des villes [Wu, 1998]. De la même manière, les automates cellulaires ont été utilisés avec succès comme modèles descriptifs avec des objectifs plus

---

<sup>9</sup> Une partition de Voronoi est une décomposition particulière d'un espace métrique déterminée par les distances à un ensemble discret d'objets de l'espace, en général un ensemble discret de points. Pour des utilisations en modélisation urbaine par automate cellulaire, voir les références qui suivent dans le texte.

théoriques, par exemple l'exploration de la croissance urbaine spontanée [Batty, 1998], ou l'étude des processus de ségrégation et de polarisation (Portugali, 2000; Portugali *et al.*, 1997). Le principal avantage des automates cellulaires est de donner une représentation explicite de l'espace et des voisinages. Des formes complexes de cellules sont envisageables, par exemple, par le couplage avec des systèmes d'information géographique (Batty *et al.* 1999 ; Coquillard & Hill, 1997 ;Kirtland *et al.*, 1994). Toutefois, les automates cellulaires sont conçus dans une approche ascendante stricte où seulement les interactions au niveau microscopique interviennent. Les interactions entre différents niveaux ne sont pas considérées ou sont considérées de façon statique, sous forme de contraintes globales qui font office de régulateurs du système. Une autre faiblesse des automates cellulaires est qu'ils ne sont pas adaptés à représenter des entités qui se déplacent. Même s'il existe quelques exemples d'automates cellulaires avec des cellules mobiles (Benenson 1998; Schofisch & Haderler 1996; Wahle *et al.* 2001), l'incapacité originale de représenter des automates mobiles a conduit les géographes à s'intéresser plutôt aux systèmes multi-agents. Cette tendance est très importante en géographie urbaine, où les automates cellulaires sont insuffisants pour modéliser des entités mobiles telles que des piétons, des ménages qui déménagent, etc. Le plus souvent les automates cellulaires sont utilisés pour représenter la « couche spatiale » dans des modèles hybrides où la « couche mobile » est normalement modélisée par des systèmes multi-agents. Ce type de modèles hybrides est connu sous le nom de modèles de Géosimulation [Benenson & Torrens 2004a ; 2004b]. Un grand nombre d'exemples de ce type de modèle hybride peuvent être trouvés dans le site web : <http://www.geosimulation.org/>

#### 2.2.2.4. Les systèmes multi-agents

Depuis le milieu des années 1990, les systèmes multi-agents (SMA) développés notamment par Feber et Wooldridge [Ferber, 1995, 1999 ; Wooldridge, 2002], connaissent de nombreux développements dans les sciences sociales. Ils offrent la possibilité de représenter directement les comportements et même les interactions entre individus pour expliquer et parfois prévoir l'évolution de phénomènes complexes. Il s'agit d'expliquer comment des dynamiques et des phénomènes complexes à un niveau agrégé d'observation peuvent émerger d'interactions qui se produisent à un niveau local. Dans cette perspective, les systèmes multi-agents représentent des systèmes auto-organisés. Ils permettent aussi d'appréhender des systèmes complexes avec un grand nombre de composants, leurs interactions et leurs différents niveaux de représentation, qu'ils soient prédéterminés ou émergents. Nous présentons ici une série de définitions essentielles.



### Agent :

Un agent est un type évolué d'automate. La différence principale avec les automates basiques est que les agents représentent normalement des entités qui prennent des décisions de manière autonome (Epstein 1999 ; Kohler 2000). D'un point de vue informatique, un agent décrit une abstraction informatique similaire à la programmation orientée objet, sauf qu'au lieu d'être défini en termes de méthodes et d'attributs, un agent est défini en termes de comportements. Un agent est un programme informatique auto-contenu qui peut contrôler ses propres actions à partir de ses perceptions de son environnement et de son état interne [Huhns et Singh 1998]. Les principales caractéristiques qui définissent le comportement d'un agent sont (prises de [Gilbert & Troitzsch, 1999]):

- l'autonomie : un agent agit sans contrôle externe direct de ses actions ou de ses états internes.
- la réactivité : un agent est capable de percevoir son environnement (qui peut être le monde physique, un monde virtuel ou simulé, ainsi que les autres agents) et d'agir en réponse à l'état courant de cet environnement.
- la capacité sociale : les agents communiquent et interagissent avec les autres agents.
- la pro-activité : en plus de réagir à l'état de son environnement, un agent peut être capable de prendre des initiatives en vue d'un but précis.

### Système multi-agents :

On parle de systèmes multi-agents (SMA) lorsque l'on considère plusieurs agents qui communiquent et interagissent entre eux et qui occupent un environnement. Les SMA sont des systèmes décentralisés, où l'information et les ressources sont distribuées (chaque agent possède une partie limitée de l'information et des ressources et agit de manière indépendante). Les agents doivent interagir pour pouvoir réaliser la tâche pour laquelle ils ont été conçus. À la différence des automates cellulaires où toutes les entités (cellules) sont généralement homogènes et statiques, dans les SMA, on considère normalement des agents de différents types agissant à différents niveaux de modélisation et capables de se déplacer dans leur environnement. Dans les automates cellulaires, seules des interactions du type environnement-environnement (interactions entre cellules) sont possibles, tandis que dans les systèmes multi-agents, il existe trois types possibles d'interaction : agent-agent, agent-environnement et environnement-environnement [Barros, 2004].

### Les systèmes multi-agents en modélisation des dynamiques urbaines

Les SMA ont commencé à être utilisés en géographie à partir de la fin des années 90 (Benenson 1998; Portugali et al. 1997; Sanders et al. 1997 ; Holm et Sanders, 2007 ; Daudé,

2002). Dans le domaine des dynamiques urbaines, les SMA ont été utilisés pour reproduire différents faits spatiaux : habitat compact ou dispersé [Kohler et *al.*, 2000], hiérarchie du peuplement [Batty, 2001], formes de ségrégation spatiale (Portugali et *al.* 1997, Torrens 2001), formes d'expansion urbaine [Barros et Alves, 2003]. Parmi les modèles urbains les plus connus se trouvent le modèle SIMPOP (Bura et *al.*, 1996 ; Sanders et *al.* 1997) qui a été l'un des premiers modèles multi-agents développés pour modéliser l'émergence des réseaux urbains. Les modèles de Portugali [Portugali, 2000] et Bonnefoy [Bonnefoy, 2002] ont abordé la question de la formation des ségrégations urbaines en fonction des préférences ou des représentations des résidents. Dans beaucoup de modèles, les SMA ont été utilisés de manière combinée avec des automates cellulaires (grilles homogènes) ; dans ce cas, les agents se déplacent entre les cellules [Benenson & Torrens 2004a ; 2004b]. Souvent, les auteurs supposent que la migration des agents dépend des caractéristiques des cellules dans le voisinage (Epstein & Axtell 1996; Portugali 2000 ; Badariotti, 2002 ; Batty, 2001). Ainsi, par exemple, Portugali et Benenson [Portugali & Benenson, 1997] ont construit un SMA qui a pour objectif de reproduire la distribution des ménages dans la ville de Tel-Aviv, entre 1960 et 1995. Dans ce modèle, l'environnement est représenté par un automate cellulaire, où chaque cellule correspond à une maison ; les agents sont des ménages qui migrent à l'intérieur de la cité artificielle en fonction de leurs caractéristiques économiques et des caractéristiques des maisons.

D'autres modèles géographiques basés sur les SMA utilisent des définitions de l'espace plus proches de l'espace réel. Ils sont par exemple basés sur des systèmes d'information géographiques (Benenson et *al.*, 2003 ; Brown et *al.*, 2005; Torrens et Benenson, 2005) ou des unités du paysage (routes et points de vue) [Gimblett, 2002]. Un exemple de ce type de modèles est le projet MIRO qui est actuellement en cours de développement (Marilleau et *al.*, 2005a ; 2005b ; Banos et al, 2006a ; 2006b). Le projet MIRO a pour objectif de développer un outil de simulation des mobilités quotidiennes dans une ville. L'accent est mis dans la modélisation des activités quotidiennes individuelles. Cet outil est destiné à assister les aménageurs dans la définition de nouvelles politiques temporelles et/ou de transport. C'est un outil qui permettra de tester différents *scenarii*, en jouant sur les niveaux d'accès à la ville (rythme et horaires des services de transport, contraintes modales sur les réseaux de transport). Ce modèle est au croisement entre la *time-geography* et les systèmes multi-agents. De cette manière, ils considèrent des agents rationnels (agents VON-BDI [David, 1996]) qui sont contraints à accomplir des tâches et à organiser leur emploi du temps quotidien selon les contraintes spatio-temporelles et une hiérarchie des tâches. Dans ce système, chaque agent

représente un citoyen et les tâches représentent les activités journalières des citoyens (achats, accompagnement, domicile, loisir, restauration, travail et visites). Ces tâches sont organisées selon une hiérarchie qui permet à chaque agent de définir des priorités afin d'organiser son emploi du temps. Les tâches sont classées en plusieurs catégories : impératives (lieu et horaire fixes), flexibles (avec une grande marge de manœuvre quant au lieu et au temps de réalisation) et optionnelles (similaires au flexibles sauf qu'elles ne sont pas obligatoires). Dans MIRO, la ville est considérée comme un réseau d'unités de stockage élémentaires (les bâtiments) [Marilleau et al., 2005a] qui accueillent temporairement au cours de la journée les citoyens mobiles. La ville virtuelle est construite à partir des données réelles extraites d'un SIG. Les auteurs de ce modèle ont aussi proposé une méthodologie pour construire une population synthétique d'agents à partir d'une enquête de mobilité quotidienne [Banos et al, 2006a] : à partir des données de l'enquête, des « groupes » ont été identifiés par analyse statistique multivariée (Analyse des Correspondances Multiples et Classification ascendante hiérarchique), ensuite, des agents sont construits par un processus semi-aléatoire qui définit pour chaque agent une série de variables de mobilité et l'affecte à un des groupes résultants de l'enquête. Un certain nombre d'expérimentations ont ainsi été réalisées sur la ville de Dijon, notamment pour valider la construction de la population synthétique [Banos et al, 2006b].

La simulation des mobilités intra-urbaines à Bogotá a déjà été tentée, par le travail de Vanbergue [Vanbergue, 2000], qui propose un système multi-agent couplé à un automate cellulaire. Chaque cellule de l'automate représente un secteur du recensement de 1993 ; les cellules ont une zone d'influence et échangent des flux de population entre leurs zones d'influence. Les agents sont des ménages qui ont des préférences quant aux types d'habitats et qui essaient de se déplacer quand ils ne sont pas satisfaits. La satisfaction dépend de la distribution de la population dans le secteur. L'offre de logements est calculée en fonction d'une capacité maximale fixe de secteurs urbains. Les agents « ménages » sont réactifs (ils recalculent leur satisfaction lors des modifications des caractéristiques de leur secteur urbain) et rationnels (ils essaient de maximiser leur niveau de satisfaction et déménagent en conséquence). Lalage [Lalage, 2001] étend ce travail en s'intéressant à la représentation des mobilités par des systèmes d'information géographique (SIG). La satisfaction dans ce travail est fonction du *standing* du logement, des critères économiques (avoir un « bon » travail) et familiaux (préserver l'unité familiale) ainsi que des caractéristiques du secteur urbain (composition de la population ; infrastructures : bibliothèques, centres éducatifs, hôpitaux, églises ; sécurité ; accès aux services publics ; usages). Lalage considère deux niveaux de modélisation en définissant deux types d'agents différents : des agents « individuels »

représentant les ménages et des agents « collectifs » qui représentent des institutions tels que celles de l'aménagement et de la planification urbaine, ainsi que la police. Ces agents collectifs modifient le secteur urbain selon des politiques pré-établies.

### **2.3. Conclusion et proposition d'une approche multi-niveaux intermédiaire**

Nous avons présenté plusieurs types de modèles utilisés dans la modélisation des dynamiques urbaines. Dans le cas de la modélisation de systèmes urbains (mais aussi spatiaux en général), on a relevé la nécessité de considérer :

- 1) au moins deux niveaux de modélisation ;
- 2) l'espace géographique de manière explicite ;
- 3) les interactions entre entités au niveau individuel ;
- 4) les effets des évolutions globales sur le niveau individuel ;
- 5) l'action des entités individuelles sur les variables globales.

Historiquement, plusieurs types de modélisation ont été proposés pour relever le défi imposé par l'étude de dynamiques sociales et en particulier les dynamiques urbaines. A partir des années 1960, avec l'essor des modèles de microsimulation, on a commencé à considérer au moins deux niveaux de modélisation. Ces modèles ont été conçus pour étudier l'influence des changements globaux sur une population. L'introduction de l'espace et des interactions individuelles a été un peu plus tardive, et plusieurs solutions ont été proposées par les modèles de type ascendant. Toutefois, une solution complète qui tienne compte des cinq considérations présentées n'est pas encore clairement définie.

Nous avons constaté plusieurs inconvénients des approches multi-niveaux classiques. Dans le cas de l'approche descendante, il est très facile de tomber dans la simplification extrême et de sur-évaluer l'effet des variables globales sur l'évolution d'une population. C'est peut-être la raison pour laquelle les modèles de microsimulation classique donnent des résultats très peu exacts [Boman & Holm, 2005]. De même, leur pouvoir prédictif (qui était le but de leur conception) est limité. Dans le cas de l'approche ascendante, le risque est de tomber rapidement dans des simulations très complexes dont les résultats sont très difficiles à interpréter et de contredire ainsi un des piliers de la modélisation, c'est-à-dire : la simplification voulue mais aussi nécessaire de la réalité. Il s'agit donc de trouver un compromis entre les deux approches : pas de simplification excessive avec une vision trop

globale mais pas non plus de trop grande complexité avec une vision trop individualisée. D'un autre côté, on observe aussi, que lorsque l'on considère le niveau microscopique dans la modélisation, il faut introduire une dose importante de règles stochastiques. L'introduction de ces règles stochastiques est nécessaire car on suppose qu'il n'existe pas de séquences ordonnées dans les interactions ou évolutions individuelles : il faut éviter d'actualiser dans un ordre préétabli les états des agents ou micro-entités. Toutefois, l'utilisation excessive de règles stochastiques nuit généralement à la reproductibilité des résultats des simulations. Nous ne nions pas la nécessité de rompre avec des schémas ordonnés d'actualisation des micro-entités, mais voulons attirer l'attention sur les effets négatifs d'une utilisation excessive des règles stochastiques. Il existe des alternatives à l'utilisation des règles stochastiques, comme les architectures basées sur des tableaux noirs (blackboards systems [Corkill, 1991]), mais ce type de stratégies suppose une augmentation importante de l'utilisation des ressources informatiques pour pouvoir garder les états futurs. Cela peut aussi complexifier le modèle.

Dans notre cas d'étude, nous nous intéressons à la modélisation des processus de mobilité résidentielle dans une ville afin de pouvoir reproduire la distribution de la population sur une période de temps donnée. Notre modèle fait suite au modèle dynamique développé par Piron, Mullon, Dureau [Piron et al, 2007], à l'échelle de la ville. Basé sur un modèle économique du marché du logement et formalisé dans le cadre de la théorie des jeux, leur modèle s'appuie également sur des logiques sociales et démographiques de la mobilité des ménages. Il évalue pour cela l'évolution et la redistribution des ménages qui changent de logement à l'intérieur de la ville en couplant plusieurs dynamiques : une dynamique démographique du cycle de vie des ménages, une dynamique de la mobilité sociale, une dynamique du parc de logements et une dynamique économique du marché du logement. Ils proposent de se placer à un niveau intermédiaire de l'organisation socio-résidentielle de la ville tout en fixant comme entités de modélisation le ménage et le logement, définis au travers de typologies éprouvées et localisées. Ceci conduit à définir d'une part des groupes sociaux pertinents, produits du cycle de vie des ménages et de leur hiérarchie sociale donnée par l'activité et le niveau d'éducation et d'autre part, des types de logements représentatifs du parc de Bogota. L'objectif est d'explicitier et de tester les hypothèses générales sur les causes des recompositions des quartiers en tenant compte simultanément des choix résidentiels d'ordre social (préférence d'un groupe social pour un type de logement) et d'ordre spatial (préférence pour un lieu donné). Il s'agit d'identifier les facteurs déterminants de la mobilité des groupes sociaux et des transformations des quartiers, d'expliquer les différenciations spatiales autour de

comportements résidentiels très typés et de représenter ainsi les tendances d'évolution à l'échelle de la ville.

Dans la continuité de ce modèle et en tenant compte des considérations présentées ci-dessus, nous proposons d'une part la réduction de la complexité des simulations non pas en considérant les interactions au niveau microscopique comme moteur des mobilités résidentielles, mais les interactions d'entités abstraites localisées à un niveau intermédiaire (niveau mésoscopique). Ces entités abstraites, groupes sociaux et types d'habitats, offrent une vision synthétique mais pas schématique de la population et du parc du logement. Le nombre d'entités considérées est ainsi réduit et les interactions simplifiées. Mais par rapport au modèle de [Piron et al., 2007], nous souhaitons reproduire l'évolution de ces entités. à partir de l'évolution de la population et du parc du logement. Pour cela, nous proposons l'utilisation de règles globales d'évolution basées sur des quotas de ménages et de logements.

D'un autre côté, à partir des recherches bibliographiques réalisées, il semble que le fait de considérer des structures abstraites évolutives, comme des groupes sociaux ou des types d'habitats, n'a pas été traité. En effet, dans les approches ascendantes, on observe deux tendances :

- 1) considérer explicitement un niveau microscopique avec des micro-entités d'un ou plusieurs types ;
- 2) considérer explicitement un niveau microscopique (individus, ménages, logements) et un niveau mésoscopique avec des méso-entités plus ou moins concrètes comme des firmes ou des institutions.

Dans le premier cas, on définit les interactions au niveau individuel, et l'on observe les résultats au niveau global (distributions de population, usage du sol, etc.). Dans le deuxième cas, les méso-entités sont prédéfinies, et n'évoluent pas ou évoluent d'une manière très restreinte. De plus, seulement quelques interactions entre les micro-entités et les méso-entités sont considérées. Dans notre cas, nous nous intéressons à la modélisation de méso-entités qui sont des abstractions de la réalité. Elles doivent se constituer au cours des simulations, et doivent être capables d'interagir entre elles, d'évoluer et d'interagir avec les micro-entités. Dans le chapitre suivant, nous donnerons quelques pistes par rapport à la formation et à l'évolution automatique de ce type de méso-entités.

## **Chapitre III : Les mécanismes de formation de groupes**

Dans la littérature classique, la formation de groupes d'individus semblables est communément associée à des méthodes de classification de données. Ces méthodes issues de l'analyse statistique ont été très étudiées et on dénombre actuellement un nombre très important d'algorithmes et de familles d'algorithmes de classification. Le problème de la formation de groupes a été aussi traité par des méthodes de simulation, notamment des simulations multi-agents. Dans ce cas, il s'agit généralement de modèles de simulation qui servent à étudier des problèmes physiques ou biologiques. Dans ces modèles, on considère d'une part la formation automatique d'agents composés d'entités élémentaires et d'autre part l'interaction entre les agents composés et entre les agents composés et les entités. Selon le problème, les agents composés peuvent représenter des organismes, des molécules ou des structures mécaniques.

Le problème de la formation automatique de groupes nous intéresse tout particulièrement, car c'est la base de la constitution de groupes sociaux et de types d'habitats dans notre système. D'un autre côté l'interaction entre groupes sociaux et types d'habitats est pour nous la base de l'évolution de la distribution de la population. Pour cette raison, nous nous intéressons aux interactions avec des agents composés qui sont formés de manière automatique.

Nous considérons le problème de constitution automatique de groupes sociaux et de types d'habitats comme un problème de classification de données évolutives. Les données à classifier sont des données socio-démographiques décrivant la population et le parc de logement d'une ville. La tâche consiste à constituer des groupes de ménages et de logements et à leur permettre d'évoluer. Notre but est de faire interagir les groupes résultants de la classification avec un mécanisme de mobilité au niveau de groupes. À fin d'atteindre notre but, nous devons considérer un algorithme de classification totalement automatique : d'une part, le nombre de classes et leur structure évoluent sur une simulation ; d'autre part, afin d'avoir une simulation continue sur une période de temps, l'intervention d'un humain n'est pas autorisée (par exemple, pour choisir le meilleur ensemble parmi plusieurs proposés par une classification). L'algorithme considéré doit alors déterminer le nombre de classes et doit s'adapter à l'évolution des données. Tout d'abord, nous présentons différents algorithmes de

classification, mettant l'accent sur les algorithmes totalement automatiques. Dans la deuxième partie de ce chapitre, nous présentons différents modèles de formation et interaction d'agents composés.

### 3.1. La classification automatique

La classification automatique a pour objectif de « trouver la structure » d'un ensemble de données du même type. Les plus souvent les données représentent des objets ou des individus décrits par un certain nombre de variables ou de caractères [Lebart et *al.*, 2006]. La structure de l'ensemble des données est présentée sous forme de partition ou de hiérarchie de partitions. Chaque donnée est assignée à une *classe* dans la partition ou, dans le cas d'une hiérarchie de partitions, à plusieurs *classes*. On suppose que la structure existe et l'objectif est donc de la rendre évidente.

Les données utilisées dans une classification automatique sont recueillies et pré-traitées avant de les classifier. Le recueil des données est normalement une tâche réalisée directement sur le terrain. Il s'agit par exemple des enquêtes sur une population d'individus, ou des mesures sur une expérience de laboratoire. Le prétraitement est un processus d'uniformisation des données : les données sont « représentées » sous une forme standard, par exemple sous forme de tableau de modalités. Il faut noter qu'on ne peut parler directement de modalités que pour les variables nominales : lorsqu'il s'agit d'une variable continue, il faut d'abord la découper en modalités. Une modalité est la valeur d'une caractéristique : par exemple, la caractéristique *genre* dans le cas d'une enquête a les modalités « *homme* » et « *femme* ». Dans un tableau de modalités, une ligne représente une donnée et chaque colonne représente une modalité. Une des formes standard de représentation des données est la forme disjonctive complète. Dans la forme disjonctive complète, chaque donnée  $d$  est représentée par un vecteur binaire à  $p$  (où  $p$  est le nombre de modalités) composantes :

Par exemple si la seule caractéristique considérée dans une enquête de population est le *genre*, une femme serait représentée par le vecteur  $[1\ 0]$  et un homme serait représenté par le vecteur  $[0\ 1]$ . Nous adoptons la forme disjonctive complète pour représenter les données dans notre système.

Le processus de classification automatique comporte deux étapes essentielles :



- Le partitionnement
- L'interprétation et la validation des partitions

Le partitionnement est la phase de mise en évidence de la structure qui décrit l'ensemble des données. Il existe un très grand nombre d'algorithmes de partitionnement. Classiquement, ils se classifient en algorithmes agglomératifs et en algorithmes hiérarchiques. La phase de validation et d'interprétation des partitions est généralement un processus complexe et difficile à rendre automatique.

Dans la suite de notre travail, nous présentons certains des algorithmes classiques de partitionnement. Puis, nous présentons des approches de partitionnement par les systèmes multi-agents et finalement quelques techniques de validation et d'interprétation automatique des résultats d'un partitionnement.

### 3.1.1. Méthodes classiques de partitionnement (clustering)

L'éventail d'algorithmes de partitionnement est importante dont un bon aperçu est donné dans (Nakache et Confais, 2005) . Nous présentons quelques-uns des algorithmes les plus connus, que nous considérons comme représentatifs de leur catégorie. Les algorithmes de partitionnement ont en commun deux objectifs : maximiser la compacité à l'intérieur des partitions et maximiser la différenciation entre partitions. Le premier objectif consiste à trouver la partition la plus compacte possible, c'est-à-dire la partition pour laquelle la différence entre les éléments dans une même classe est minimale. Le deuxième principe consiste à trouver la partition où les classes sont les plus différentes les unes des autres. Ces deux principes sont illustrés dans la figure ci-dessous. Pour pouvoir évaluer la compacité et la différenciation des classes, il est nécessaire de définir une mesure de distance entre éléments et un critère d'agrégation. Plusieurs choix sont possibles, par exemple la distance euclidienne, la distance de Hamming, la distance de Khi2, etc. (Une présentation de plusieurs mesures de la distance est faite dans l'Annexe III). La compacité est souvent mesurée à partir de la variance interne d'une classe ( $\sigma_w$ ). La variance interne d'une partition est connue sous le nom de variance *intra-classes*. La différenciation entre classes est mesurée à partir de la variance entre classes ( $\sigma_b$ ), c'est-à-dire de la variance de la distance entre éléments dans des classes différentes. La variance entre classes est connue sous le nom de variance *inter-classes*. L'objectif est donc, de minimiser la variance intra-classes ( $\sigma_w$ ) et de maximiser la variance inter-classes ( $\sigma_b$ ).

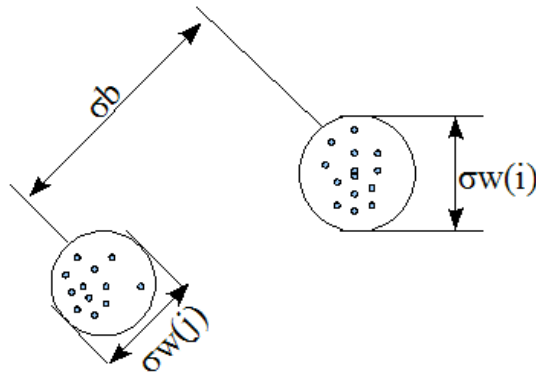


Figure 2. Illustration des principes de compacité et différenciation des partitions dans un processus de partitionnement.  $\sigma_b$  est la variance inter-classes,  $\sigma_w(i)$  est la variance intra-classe de la classe  $i$ .

Dans le partitionnement, on suppose que dans l'ensemble de données existe une structure « cachée » et qu'il est possible de la mettre en évidence. Dans la pratique, la structure trouvée dépend de l'algorithme utilisé, de son paramétrage et de la mesure de la distance utilisée. De plus, il se peut qu'il n'existe aucune structure ; dans ce cas, les résultats du partitionnement peuvent donner des partitions aberrantes.

Il existe deux types classiques d'algorithmes de partitionnement : les algorithmes agglomératifs et les algorithmes hiérarchiques. Dans la suite, nous présentons quelques exemples des algorithmes appartenant à chacune de ces catégories.

### 3.1.1.1. Algorithmes agglomératifs de partitionnement

Dans l'approche agglomérative, l'idée est de trouver des « prototypes » représentatifs de chaque classe, auxquels les données sont assignés selon leur distance au prototype. Ces prototypes sont normalement des vecteurs « moyens » des caractéristiques dans une classe. Nous présentons ici deux des familles de ces type d'algorithmes : les nuées dynamiques et les réseaux de neurones artificiels concurrents (RNA concurrents)

#### 3.1.1.1.1. Les nuées dynamiques (ou centres mobiles ou k-means)

Cette famille d'algorithmes a été étudiée dans un cadre formel par Diday [Diday, 1979]. Dans cette première famille d'algorithmes agglomératifs, on définit un ensemble de noyaux  $N_0$  à partir desquels on définit une partition  $P_0$  par l'intermédiaire d'une fonction  $f$ . Les noyaux et les partitions sont ensuite actualisés de manière à minimiser une mesure d'erreur donnée par une fonction  $H$ . Le macro-algorithme de ce type de méthodes est présenté ci-dessous.  $H$  est une fonction qui mesure l'adéquation entre  $P$  et  $N$ . Dans l'algorithme, la valeur rendue par  $H$  diminue jusqu'à atteindre une valeur minimale. Cette valeur minimale est la condition d'arrêt de l'algorithme.

DEBUT

1. Choisir un ensemble de noyaux initiaux  $N_0$ .
2. Dédire par  $f$  la partition initiale :  $P_0 = f(N_0)$ .
3.  $i = 0$
4.  $i = i + 1$
5. Déterminer un nouvel ensemble de noyaux  $N_i$  tel que :

$$H(P_{i-1}, N_i) \leq H(P_{i-1}, N_{i-1})$$

6. Déterminer la nouvelle partition  $P_i = f(N_i)$  tel que :

$$H(P_i, N_i) \leq H(P_{i-1}, N_i)$$

7. Si  $H(P_i, N_i) < H(P_{i-1}, N_i)$  revenir à 4

FIN

Encadré 1. Macro-algorithme de partitionnement par la méthode de nuées dynamiques

Deux des implémentations les plus connues de cette famille d'algorithmes sont l'algorithme des centres mobiles (Forgy, 1965 ; Thorndike, 1953 ; MacQueen, 1967 ; Ball & Hall, 1967) et l'algorithme K-means [MacQueen, 1967]. Dans ces deux algorithmes, les noyaux représentent les centres des classes. Le nombre de centres  $q$  et par conséquent le nombre de classes est connu au préalable. Les centres initiaux sont souvent choisis de manière aléatoire parmi les données à partitionner, ou construits de manière aléatoire à partir des valeurs possibles des caractéristiques décrivant les données. La fonction  $f$  assigne chaque donnée à la classe dont le centre est le plus proche de la donnée. Les nouveaux centres sont déterminés en prenant les centres de gravité de chaque classe. La mesure  $H$  d'adéquation entre les centres et les classes est normalement la variance intra-classes. On peut démontrer que la valeur de la variance intra-classe ne peut que diminuer entre des actualisations successives des centres [Lebart et al., 2006]. La mesure de la distance usuellement utilisée dans ces deux algorithmes est la distance euclidienne.

La différence fondamentale entre les algorithmes de centres mobiles et K-means est dans le processus d'actualisation des centres. Dans le premier algorithme, les centres sont actualisés après que l'on a assigné toutes les données à toutes les nouvelles classes. Dans l'algorithme K-Means, un centre est actualisé à chaque fois que l'on assigne une nouvelle donnée à la classe qu'il représente.

Les algorithmes des nuées dynamiques sont basés sur un formalisme simple. Ils sont très efficaces dans la classification des grands ensembles des données. Toutefois, ils sont très dépendants de l'ensemble des noyaux initiaux, de la méthode d'actualisation des noyaux et de la mesure de la distance utilisée. Dans les algorithmes présentés, l'initialisation des centres se fait de manière aléatoire. Ce type d'approche peut nuire à la reproductibilité des résultats des

classifications sur un même ensemble de données. Un dernier inconvénient de ces algorithmes est que le nombre de classes est connu au préalable, ce qui implique que l'on a une connaissance a priori de la structure que l'on prétend découvrir.

#### 3.1.1.1.2. Réseaux de neurones artificiels (RNA) concurrents

Une deuxième famille d'algorithmes agglomératifs de classification automatique sont les RNA concurrents. Ce sont une forme particulière de réseaux de neurones artificiels. Tout d'abord, nous faisons une présentation sommaire de ce qu'est un réseau de neurones artificiels et ensuite, nous présentons les RNA concurrents. Finalement, nous décrirons deux types de RNA concurrents : les cartes auto-organisatives de Kohonen et les *Growing Neural Gas*.

Les réseaux de neurones artificiels (RNA) sont des réseaux fortement connectés d'unités élémentaires (neurone formel) fonctionnant en parallèle. Chaque neurone formel calcule une valeur numérique en sortie sur la base des informations qu'il reçoit. Ces informations sont présentées sous forme de vecteur numérique. L'étude de RNA s'inspire en grand partie du fonctionnement du système d'apprentissage du cerveau des mammifères. Le modèle le plus simple de RNA est le perceptron. Un perceptron [Rosenblatt, 1958] est composé d'un seul neurone. Un perceptron calcule une combinaison linéaire des valeurs du vecteur d'entrée avec des poids synaptiques (voir la figure ci-dessous) ; le résultat est alors transformé par une fonction d'activation généralement non linéaire. La valeur finale obtenue est la sortie du perceptron. Les poids sont ajustables par une méthode d'apprentissage supervisée. Le principe consiste à changer les poids dans le cas où le perceptron produirait une sortie erronée pour une entrée donnée. La fonction d'activation la plus simple est la fonction de seuil : si le résultat de la combinaison linéaire est plus grand qu'un seuil donné, la sortie du perceptron est 1, autrement -1. Le perceptron est en effet un discriminateur linéaire. Il constitue un hyperplan qui partitionne l'espace de vecteurs d'entrée en deux demi-espaces  $H^+$  et  $H^-$  tel que :

$$w_1 x_1 + \dots + w_N x_N > b \rightarrow x \in H^+$$

$$w_1 x_1 + \dots + w_N x_N < b \rightarrow x \in H^-$$

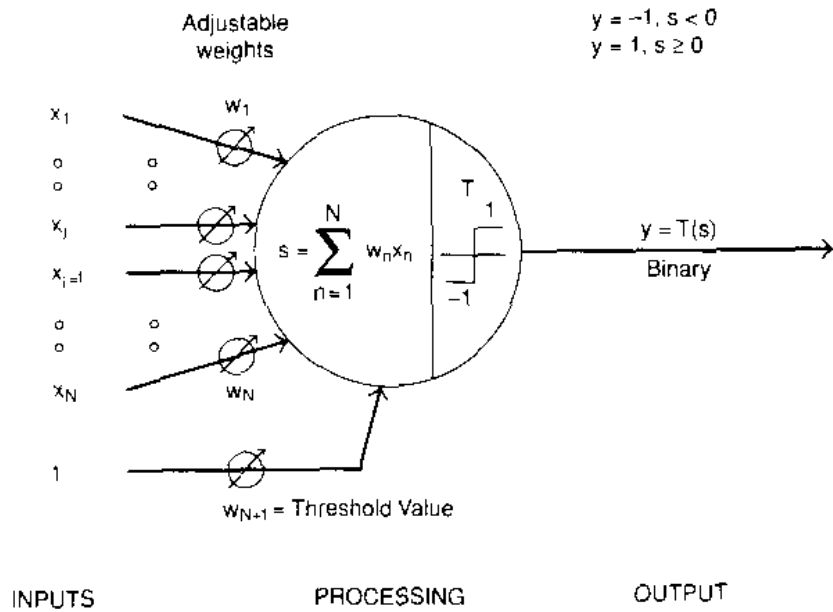


Figure 3. Le perceptron, source [Looney, 1997]

Le perceptron ne sert qu'à classer des configurations séparables linéairement [Minsky & Papert, 1988]. Pour pallier cette limitante, à la fin des années 1980, on a proposé l'idée de considérer des ensembles ou couches interconnectées contenant plusieurs perceptrons. Ces modèles sont connus sous le nom de perceptron multi-couche (MLP : Multiple Layer Perceptron voir figure ci-dessous) [Hornik et al., 1989]. En général, les méthodes d'apprentissage utilisées pour ajuster les poids synaptiques des perceptrons sont de type supervisé.

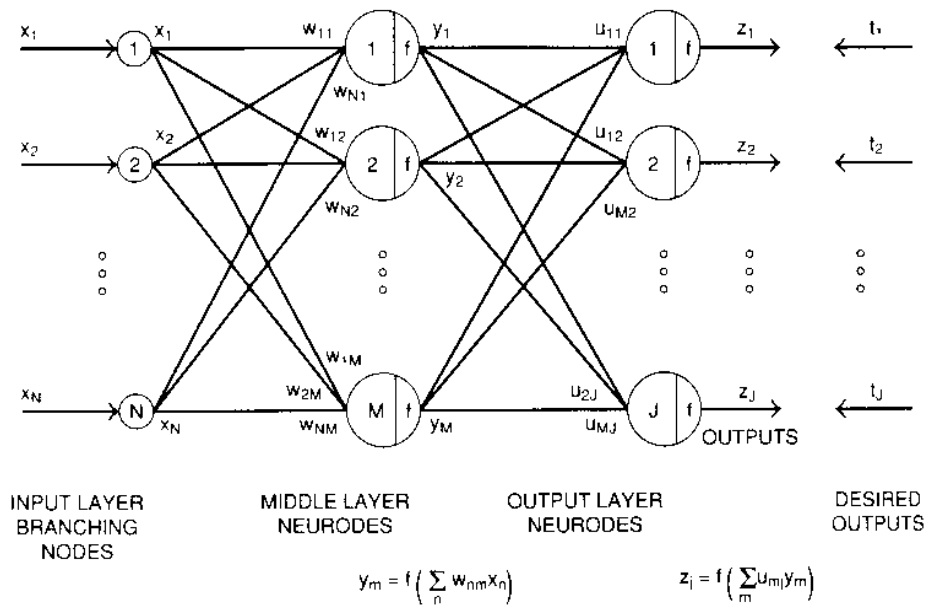


Figure 4. Exemple de perceptron multi-couche (MLP : Multi-Layered Perceptron). Le MLP dans l'exemple a une couche cachée (Middle Layer). Source [Looney, 1997]

Les RNA concurrentiels sont des réseaux à deux couches, où tous les neurones de la couche cachée sont connectés à tous les nœuds de la couche d'entrée (voir figure ci-dessus). Les vecteurs de poids des neurones dans la couche cachée sont déterminés par un processus d'apprentissage non supervisé. Ce processus est une compétition entre les neurones de la couche cachée. Un ensemble de données est présenté en entrée du réseau. À chaque fois, on détermine le neurone dont le vecteur de poids est le plus proche du vecteur d'entrée. On appelle ce neurone le « vainqueur ». Et ensuite on actualise le vecteur de poids du vainqueur. L'actualisation est effectuée selon une fonction qui dépend de la proximité du vecteur de poids au vecteur d'entrée. L'actualisation se répète sur un nombre important de cycles. Dans chaque cycle, chacune des données est présentée au réseau. Les différentes données produisent des vainqueurs différents. Cette phase d'actualisation des poids des neurones est appelé la phase d'apprentissage.

À la fin de la phase d'apprentissage, on espère qu'un seul neurone soit associé à un seul sous-ensemble de données. Parfois, certains neurones ne sont associés à aucun sous-ensemble de données, parfois plusieurs neurones sont associés à un même sous-ensemble. Le vecteur de poids d'un certain neurone est proche du vecteur moyen du sous-ensemble de données qu'il représente.

#### *3.1.1.1.2.1. Les cartes auto-organisatives de Kohonen*

Les cartes auto-organisatives de Kohonen (SOM : Self Organizing Mapping) [Kohonen, 1988] sont un type particulier de RNA concurrentiels. Dans ce type de RNA les neurones de la couche cachée sont interconnectés entre elles ce qui implique une notion de voisinage. Plusieurs configurations sont possibles : la plus simple est la carte linéaire, dans laquelle chaque neurone n'est connecté qu'à deux autres neurones. Les cartes les plus utilisées sont bidimensionnelles, dont notamment les rectangulaires (chaque cellule est connecté à quatre voisins) et les hexagonales. D'autres configurations sont possibles tels que la cylindrique ou sphérique, dans ces dernières la carte n'a pas d'extrémité.

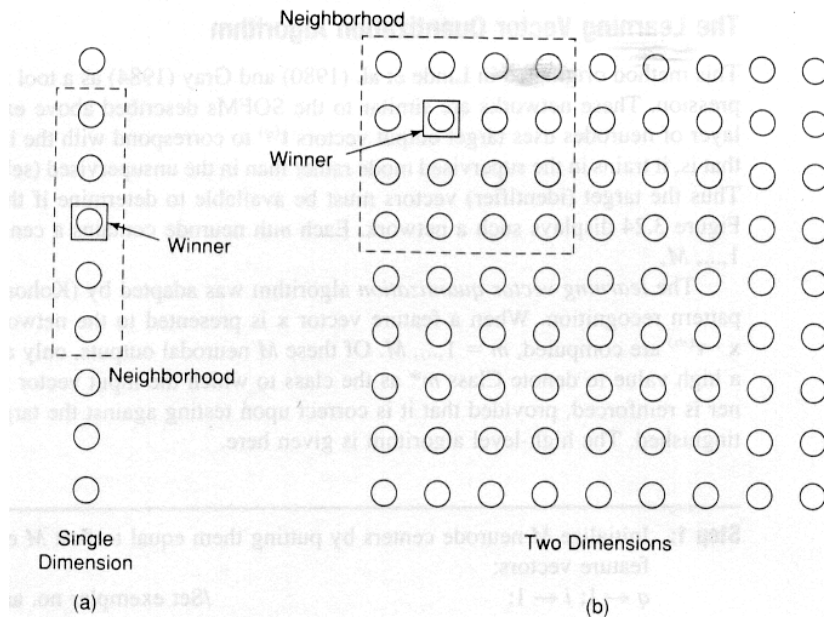


Figure 5. Exemples de configurations d'une carte auto-organisative de Kohonen, source [Looney, 1997]. Dans l'exemple on présente une carte lineaire (a) et une carte rectangulaire (b). On présente aussi le voisinage considéré dans l'actualisation des poids autour du neurone vainqueur (« winner »).

Dans la phase d'apprentissage à différence des RNA concurrentiels classiques, non seulement les poids du vainqueur sont actualisés mais aussi ceux d'un certain nombre de ses voisins (voir figure ci-dessus). L'actualisation des neurones voisins dépend de leur distance au vainqueur et de la différence entre le vecteur d'entrée et le vecteur de poids :

$$w_{i,j}(t+1) = w_{i,j}(t) + (\Theta(j, t) \times \tau(t) \times (x_i(t) - w_{i,j}(t)))$$

Où :

$t$  représente le  $t^{\text{ème}}$  cycle dans la phase d'apprentissage ;

$x_i(t)$  est la  $i^{\text{ème}}$  composante de la donnée d'entrée  $x$  ;

$w_{i,j}(t)$  est la  $i^{\text{ème}}$  composante du vecteur de poids du neurone  $j$  qu'on actualise ;

$\Theta(j, t)$  est la fonction de voisinage, qui dépend de la distance sur la grille entre le neurone qu'on actualise et le neurone vainqueur : dans sa forme la plus simple cette fonction vaut 1 pour tous les neurones les plus proches au vainqueur et 0 pour les autres, mais il est aussi usuel d'utiliser une fonction gaussienne ;

$\tau(t)$  est le coefficient d'apprentissage qui évolue de manière décroissante, il est grand au début pour accélérer l'apprentissage et petit après pour affiner les poids synaptiques.

La principale contrainte des cartes auto-organisatives est la dépendance des résultats à : 1) la topologie de la carte (la géométrie et le nombre de neurones), 2) la définition de la fonction de

voisinage, 3) les paramètres initiaux ( $\_initial$ ), 3) l'ordre de présentation des données à la carte, 4) le nombre de cycles dans la phase d'apprentissage.

#### 3.1.1.1.2.2. *Growing neural gas (GNG)*

Les GNG (Fritzke, 1994 ; Holmström, 2002) sont un autre type de RNA concurrentiels dont le nombre et les connexions de neurones dans la couche cachée évoluent au cours de la phase d'apprentissage. Au début, on commence avec un petit nombre de neurones non connectés. Des nouveaux neurones et connexions sont ajoutés successivement et des connexions sont créées ou supprimées. L'ajout de nouveaux neurones dépend des mesures locales d'erreur et du nombre d'exemples qui ont été présentés au GNG : les nouveaux neurones sont ajoutés près des neurones qui ont l'erreur cumulée la plus importante ; on ajoute un nouveau neurone lorsque le nombre d'exemples présentés est multiple d'un facteur prédéfini  $\lambda$ . Lorsque l'on présente un nouveau exemple on détermine les neurones vainqueur et le deuxième après le vainqueur, afin de créer une nouvelle connexion entre ces deux neurones. Chaque connexion a un paramètre d'âge qui est incrémenté à chaque fois que l'on présente un nouvel exemple. Une connexion est supprimée lorsque son âge atteint la valeur  $\text{âge}_{max}$ . À chaque fois que l'on présente un nouvel exemple l'erreur cumulée du neurone vainqueur est actualisée, pour les autres neurones l'erreur cumulée est diminuée d'un facteur  $\beta$ . L'algorithme est arrêté lorsque la valeur d'une mesure de performance atteint un certain seuil.

Ce type de RNA est capable de découvrir le nombre de classes. Le résultat de la classification (nombre de classes et structure) est toutefois très dépendant des valeurs des paramètres  $\beta$ ,  $\lambda$  et  $\text{âge}_{max}$ .

#### 3.1.1.1.2.3. *Limites des RNA concurrents*

Dans les RNA concurrents on découvre plusieurs problèmes: la difficulté dans le choix de la topologie du réseau, en particulier pour les cartes auto-organisatives), la complexité du paramétrage et la dépendance dans l'ordre de présentation des données. Ces choix conditionnent la qualité des résultats. Certaines solutions sont proposées par exemple pour réduire la dépendance à l'ordre de présentation de données, on peut effectuer plusieurs phases d'apprentissage avec les mêmes données avec un ordre de présentation aléatoire, par la suite on choisit le RNA qui présente les meilleurs résultats selon une mesure de performance.

La limite dans ce type de solutions est que l'introduction de l'aléatoire réduit la reproductibilité des résultats : pour un même jeu de données, les résultats ne sont pas toujours les mêmes.



### 3.1.1.2. Algorithmes hiérarchiques de partitionnement

La deuxième famille d'algorithmes de partitionnement à laquelle nous nous intéressons est la famille d'algorithmes hiérarchiques. Ce type d'algorithme ne fournit pas une partition mais une hiérarchie de partitions sous forme d'arbre contenant  $n-1$  partitions, où  $n$  est le nombre de données à classer. Cet arbre est connu sous le nom de « dendrogramme » [Lebart et al., 2006]. Nous nous intéressons à une sous-famille particulière, les *algorithmes hiérarchiques ascendants*.

#### 3.1.1.2.1. Principe de l'algorithme

Le principe de ce type d'algorithme consiste à créer, à chaque étape, une partition obtenue en agrégeant deux à deux les éléments les plus proches (voir figure ci-dessous). Un élément est soit une donnée à classer soit une classe déjà produite par l'algorithme.

Les différents algorithmes varient par rapport à la manière de considérer le nouveau couple d'éléments agrégés.

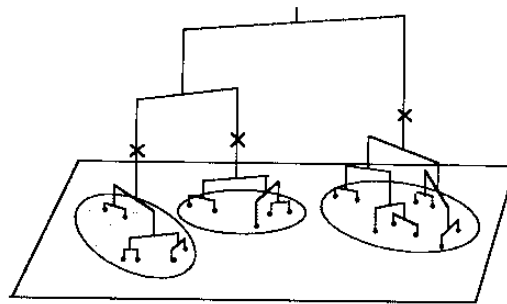


Figure 6. Dendrogramme produit par un algorithme de partitionnement hiérarchique. Source [Lebart et al., 2006]

Il faut donc définir une distance entre données, mais aussi une *stratégie de regroupement*, c'est-à-dire, il faut définir la manière de calculer la distance entre une donnée et une classe ainsi que la distance entre deux classes.

Le macro-algorithme de cette famille de méthodes est le suivant :

DEBUT

1. Les  $n$  données constituent une première partition à  $n$  classes.
2.  $nc = n$
3. On construit la matrices des distances entre les  $nc$  éléments à l'étape courante et l'on cherche les plus proches, que l'on agrège en un nouvel élément. On obtient une nouvelle partition à  $nc-1$  classes
4.  $nc = nc - 1$
5. Si  $nc > 1$  revenir à 3

FIN

Encadré 2. Macro-algorithme de partitionnement par la méthode de nuées dynamiques

### 3.1.1.2.2. Stratégies de regroupement

Il existe plusieurs possibilités de stratégie de regroupement, par exemple : le saut minimal et l'agrégation selon la variance [Lebart et al., 2006]. Dans la stratégie du saut minimal, la distance entre deux classes est la plus petite distance entre paire d'éléments dans les classes. La distance entre une donnée et une classe est la plus petite distance entre la donnée et chacun des éléments dans la classe. L'algorithme de l'agrégation selon la variance se base sur le principe de la conservation de l'*inertie totale* dans un ensemble de données. L'inertie totale est donnée par l'expression suivante :

$$\begin{array}{c}
 \textit{Inertie totale} = \underbrace{\sum_q m_q \|g_q - g\|^2}_{\textit{inertie inter - classes}} + \underbrace{\sum_q \sum_{i \in q} m_i \|x_i - g_q\|^2}_{\textit{inertie intra - classes}}
 \end{array}$$

Où,  $g$  est le centre de gravité de l'ensemble de données,  $g_q$  est le centre de gravité de la classe  $q$ ,  $x_i$  est la  $i^{\text{ème}}$  donnée dans la classe  $q$ ,  $m_q$  est le poids de la classe  $q$ ,  $m_i$  est le poids de l'élément  $i$ . Le calcul de l'inertie totale est schématisé dans la figure ci-dessous :

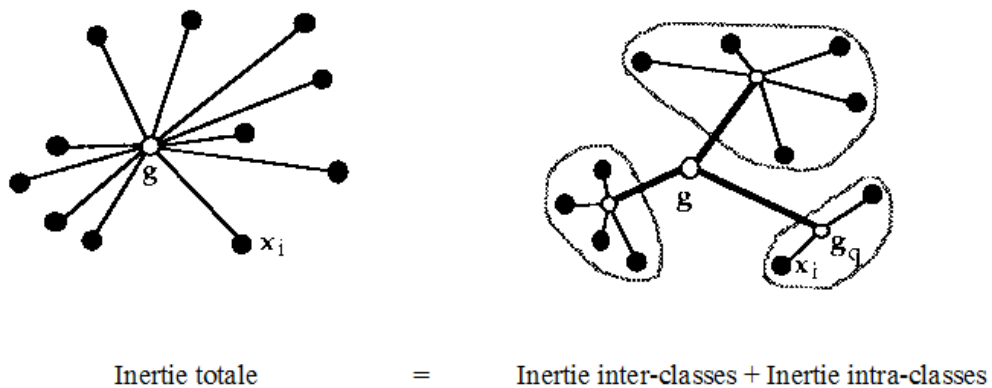


Figure 7. Inertie totale. Source [Lebart et al, 2006]

L'expression de calcul de l'inertie totale est connue sous le nom de relation d'Huygens, le premier terme de l'expression qui exprime la variance inter-classes est appelé *inertie inter-classes*, le deuxième terme qui exprime la somme des variances intra-classes est appelé *inertie intra-classes*. Le principe de conservation de l'inertie établit que la somme des inerties inter-classe et intra-classe est constante. En conséquence, maximiser l'inertie intra-classes équivaut à minimiser l'inertie inter-classes. Le principe de l'algorithme d'agrégation selon la variance consiste à rechercher à chaque étape une partition telle que la variance interne de chaque classe soit minimale et que par conséquent la variance entre les classes soit maximale. Il s'agit donc d'agréger à chaque étape les deux éléments qui respectent cette contrainte. Dans la première étape de l'algorithme ( $n$  classes, une classe par donnée), l'inertie intra-classes est

nulle et l'inertie inter-classes est égale à l'inertie totale. Dans l'étape finale (1 classe contenant toutes les données), l'inertie inter-classes est nulle et l'inertie intra-classes est égale à l'inertie totale. Pour plus de détails sur cette stratégie nous conseillons la lecture de [Lebart et al., 2006].

### 3.1.1.2.3. Choix de la meilleure partition

Dans le dendrogramme produit par une partition hiérarchique, on a un nombre  $n-1$  ( $n$  est le nombre de données à classer) de partitions possibles. Pour avoir une unique partition, il faut choisir la meilleure de ces partitions selon un critère de qualité : il s'agit de faire une *coupure* à un certain niveau de l'arbre (voir figure ci-dessous). Le choix du niveau de la coupure est normalement fait par inspection visuelle de l'arbre, il s'agit de trouver le niveau pour lequel les éléments au-dessus de la coupure sont éloignés et les éléments au-dessous de la coupure sont proches. Dans la pratique, le choix du niveau de la coupure peut être très difficile et il peut y avoir plusieurs coupures possibles.



Figure 8. Coupure d'un dendrogramme. Source [Lebart et al., 2006]

Les algorithmes hiérarchiques ont l'avantage d'être des algorithmes déterministes. Toutefois, la construction de la hiérarchie de partitions est dépendante du choix du critère d'agrégation et du calcul de la distance choisie. D'autre part, le nombre d'opérations réalisées est très important. L'inconvénient le plus important concerne le choix de la meilleure partition : le choix du niveau de la coupure est difficile à rendre automatique.

### **3.1.2. Méthodes de partitionnement par systèmes multi-agents**

Dans cette partie, nous présentons quelques algorithmes qui utilisent des agents réactifs [Drogoul et Ferber, 1994] pour la classification automatique de données. Une partie de ces algorithmes s'inspirent du comportement des insectes sociaux et considèrent des agents qui

trient des données. D'autres algorithmes s'inspirent des comportements des vols d'oiseaux et des mouvements des bandes de poissons. Ces derniers considèrent chaque donnée comme un agent qui se déplace dans un environnement dépendant des interactions avec les autres agents. Les algorithmes présentés dans cette partie se basent sur des mécanismes de perception et d'interactions locales qui permettent de former des regroupements de données semblables.

#### 3.1.2.1. Partitionnement par colonies de fourmis

Les algorithmes de colonies de fourmis (ACO : Ant Colony Optimisation) introduits par (Colomni et *al.* 1991 ; Dorigo, 1992) sont originellement des algorithmes d'optimisation : ils permettent de trouver une solution optimale à un problème parmi un grand nombre de solutions possibles. Ces algorithmes ont été récemment étendus pour résoudre des problèmes de classification automatique des données (Alexandrov, 2000 ; Machnik, 2006). Ces algorithmes s'inspirent du comportement collectif des fourmis pour trouver des chemins à partir du nid vers des sources de nourriture. Ce comportement collectif se base sur le principe de stigmergie [Grassé, 1959], c'est-à-dire, la communication indirecte entre les fourmis par la modification de leur environnement. En effet, les fourmis sont capables de se coordonner par le dépôt et la détection de phéromones. Dans le cas de la recherche de nourriture, les fourmis se déplacent initialement de manière aléatoire, mais lorsqu'une fourmi trouve de la nourriture, elle rentre au nid en marquant chimiquement le chemin par dépôt de phéromones. Petit à petit, les autres fourmis détectent ce chemin et le parcourent en le renforçant par le dépôt de nouvelles phéromones. On crée ainsi une boucle de rétroaction : de plus en plus de fourmis sont attirées vers le chemin créé qui est alors de plus en plus renforcé. La participation de l'environnement est très importante, non seulement comme espace porteur des phéromones, mais aussi comme agent de contrôle de la durée du chemin. En effet, les phéromones se dissipent au bout d'un certain temps. Lorsqu'il n'y a plus de nourriture au bout du chemin, le chemin disparaît.

Dans le cas général, la recherche de nourriture est adaptée à la recherche de données ou d'objets semblables. Ces algorithmes permettent de construire des liens (les chemins) entre objets semblables. Par exemple, dans le travail présenté dans [Machnik, 2006], l'objectif est de trouver les plus courts chemins entre documents web semblables, en créant des classes de documents interconnectés.

#### 3.1.2.2. Algorithmes basés sur le tri de fourmis

Un autre comportement des fourmis a inspiré le développement d'algorithmes de classification automatique des données. C'est le tri du couvain, observé dans certaines espèces de fourmis. Ce comportement s'adapte plus naturellement à la classification de données. Il a donc inspiré beaucoup plus de travaux que le mécanisme de recherche de nourriture présenté dans la section précédente. Les premiers travaux datent du début des années 1990 [Deneubourg et *al.*, 1990]. Ces premiers travaux se limitent au tri d'objets dans un plan (espace d'interaction) par des fourmis artificielles : tous les objets considérés sont du même type, l'objectif étant de rapprocher des objets déjà proches en formant des amas plus denses. Ces travaux ont été adaptés par [Lumer & Faieta, 1994] pour faire de la classification de données. Le principe est très simple : les données sont disposées de manière aléatoire dans une grille à 2 dimensions ; les fourmis artificielles peuvent « saisir » les données et les déposer dans des cases où la densité d'objets similaires est élevée. Chaque fourmi est localisée dans une case de la grille, et elle a une vision locale (voisinage de Moore). Si elle trouve des données dans son voisinage, elle les ramasse selon une probabilité qui dépend de la similarité entre les données du voisinage : en règle générale, plus une donnée est différente des données dans le voisinage, plus la probabilité que la fourmi la ramasse est grande (une fourmi ne peut ramasser qu'une donnée à la fois). Une fois que la fourmi a saisi la donnée, elle se déplace de manière aléatoire dans la grille afin de trouver la case la plus adaptée pour déposer la donnée. Le dépôt de la donnée dépend aussi d'une probabilité : plus les données dans le voisinage de la fourmi ressemblent à la donnée qu'elle porte, plus la probabilité de déposer la donnée est grande.

L'algorithme original a été étendu à d'autres types de problèmes comme le partitionnement de graphes [Kuntz et *al.*, 1997] ou la classification de sessions web [Abraham & Ramos, 2003]. Plusieurs adaptations de l'algorithme ont été proposées. Par exemple, un algorithme hybride entre le tri de fourmis et l'algorithme K-means [Monmarché et *al.*, 1999] qui améliore la performance en termes de temps et de précision. Dans [Labroche et *al.*, 2002], les auteurs proposent un algorithme de tri qui fait intervenir « l'odeur coloniale », c'est-à-dire un identificateur chimique similaire aux phéromones. L'odeur est en effet une représentation de la donnée moyenne de chaque classe. Dans cet algorithme, les fourmis se spécialisent pour trouver uniquement des données avec une certaine odeur et rejettent les données avec des odeurs non identifiées.

### 3.1.2.3. Algorithmes basés sur le déplacement de particules (la compétition spatiale)

Il s'agit d'une autre famille d'algorithmes de classification automatique où chaque donnée est représentée par une particule. L'algorithme de classification est basé sur des règles d'attraction-répulsion entre données. Les particules se déplacent dans un espace à deux ou trois dimensions. Les déplacements sont le résultat des interactions locales entre les particules et le résultat est la formation des amas de particules semblables qui se déplacent ensemble à une même vitesse et avec une même direction. Les interactions locales sont en effet basées sur le calcul de la similarité entre données. L'inspiration initiale de ce type d'algorithmes est l'étude du vol des oiseaux et du déplacement de bandes de poissons. Dans ces deux cas, des formes complexes peuvent se produire à partir des déplacements individuels sans aucune sorte de contrôle centralisé. Les premières modélisations informatiques de ces phénomènes sont dues à [Reynolds, 1987]. Dans ses travaux, Reynolds s'est intéressé à la reproduction réaliste des mouvements de foules d'individus. Ces travaux ont été notamment utilisés dans l'industrie du cinéma. L'étude des déplacements de bandes de poissons et de vols d'oiseaux a donné aussi naissance à une famille d'algorithmes d'optimisation : les algorithmes PSO (Particle Swarm Optimisation) [Kennedy & Eberhart, 1995].

Les premiers travaux de classification automatique basés sur les déplacements des particules sont dûs à [Proctor & Winter, 1998]. Dans ces premiers travaux, les agents (particules) se déplacent dans un espace bidimensionnel. Leur position initiale est définie de manière aléatoire. Les mouvements des agents dépendent des autres agents perçus dans le voisinage et des similarités entre les données qu'ils représentent. Le comportement local tend à former globalement des groupes d'agents qui se déplacent de manière cohérente (les agents dans un même groupe se déplacent à la même vitesse et dans la même direction). L'algorithme original a été amélioré par l'introduction de nouveaux paramètres [Monmarché et *al.*, 2002] notamment une distance idéale entre agents qui dépend de la similarité entre les données qu'ils représentent et un critère d'arrêt basé sur une mesure de l'entropie globale du nuage d'agents. Un cas spécifique de ce type d'algorithmes est proposé dans le système OSCAR (Outil de Simulation Comportementale par Attraction-Répulsion) [Renault, 2001]. Renault s'intéresse à la visualisation dynamique des flux de données. Elle propose un système dynamique de formation des groupes à partir des flux des données. Dans ce système, les données arrivants sont représentées par des agents réactifs qui habitent un environnement. Ces agents déposent des signaux qui les identifient et se déplacent dans l'environnement en fonction des gradients de répulsion-attraction générés par les signaux déposés par les autres. Ainsi les agents similaires se rapprochent et forment des groupes tout en s'éloignant des agents qui ne leur

ressemblent pas. Une hiérarchie des groupes est constituée où chaque groupe compte avec son propre environnement dans lequel habitent et agissent les agents qui le constituent.

### **3.1.3. Méthodes de validation et interprétation automatique d'une partition**

Lorsque l'on produit une partition par une méthode de classification automatique, deux questions doivent être résolues : la structure trouvée représente-t-elle aisément les données ? Quel est l'interprétation de cette structure ? Dans cette partie, nous présentons quelques mécanismes automatiques de validation et interprétation qui permettent de répondre en partie à ces deux questions.

#### 3.1.3.1. Validation des partitions

La validation d'une partition est un processus complexe qui a pour objectif de mesurer sa qualité. Il existe trois formes de validation :

- La validation externe qui mesure le degré avec lequel la partition confirme des informations connues a priori sur les données ;
- La validation interne qui mesure la stabilité des résultats obtenus.
- La validation relative qui compare plusieurs partitions et permet de décider laquelle représente le mieux un ensemble de données.

Dans le premier type de validation, on compte avec des informations sur l'ensemble des données à classer. Par exemple, les résultats d'autres méthodes de classification ou les résultats des classifications sur des échantillons. Il s'agit dans ce cas de comparer l'information donnée par la nouvelle partition avec les informations connues. Dans le deuxième cas, on teste la stabilité (reproductibilité) de la partition par rapport aux modifications dans l'ensemble de données à classer. Il s'agit normalement de techniques basées sur le ré-échantillonnage sur l'ensemble de données et sur l'introduction de bruit. Finalement le troisième type de validation permet de comparer plusieurs partitions selon la valeur d'un indice statistique qui mesure la qualité de chaque partition. À la suite, nous nous intéressons à ce dernier type de validation.

La validation relative d'une partition se base sur le calcul d'un indice qui mesure la qualité de la partition par rapport aux deux objectifs de la classification automatique : la minimisation de la variance intra-classes et la maximisation de la variance inter-classes. L'indice met en relation les variances intra et inter classes. Le calcul de l'indice permet de noter plusieurs partitions et de choisir celle qui s'ajuste le mieux aux données. Une description très complète

d'un nombre important de ces indices est faite dans [Youness, 2004] et [Milligan & Cooper, 1985]. Milligan et Cooper ont étudié 30 de ces indices et ont trouvé que les deux meilleurs sont l'indice F-Maximum et l'indice de Davis-Bouldin. Nous présentons ces deux indices ci-après.

#### 3.1.3.1.1. L'Indice F-Maximum

L'indice F-Maximum sur une partition  $P$  est donné par l'équation ci-dessous. Le calcul de cet indice tient compte de la relation entre l'inertie intra-classes et l'inertie inter-classes de la partition ainsi que de la relation entre le nombre  $n$  de données à classer et le nombre de classes  $k$  trouvées. Une bonne partition est celle qui maximise cet indice.

$$FMaximum(P) = \frac{InertieInterClasses(P)}{InertieIntraClasses(P)} \times \frac{n-k}{k-1}$$

Comme on peut le déduire de l'expression ci-dessus, l'indice F-Maximum est maximisé quand la valeur de l'inertie intra-classes est minimisée et quand la valeur de l'inertie inter-classes est maximisée. Cet indice pénalise les partitions avec un nombre trop grand de classes (pour  $k$  proche de  $n$  l'indice tend vers 0).

#### 3.1.3.1.2. L'Indice de Davis-Bouldin

Dans ce cas, les meilleures partitions sont celles qui minimisent l'indice de Davis-Bouldin : une valeur petite de l'indice veut dire que les classes sont compactes et que les centres des classes sont très éloignés les un des autres. La valeur de cet indice pour une partition  $P$  avec  $k$  classes est donné par l'expression ci-dessous :

$$DavisBouldin(P) = \frac{1}{k} \times \sum_{j=0}^k R_j(P)$$

Où :

$$R_j(P) = \max\left(\frac{S_j + S_i}{T_{ij}}\right) \quad \forall i \neq j$$

$S_j^2$  est l'écart-type de la distance entre chaque donnée et le centre de la classe  $j$ .  $T_{ij}$  est la distance entre les centres des classes  $i$  et  $j$ .

#### 3.1.3.2. Interprétation automatique d'une partition

L'interprétation automatique des classes dans une partition se base généralement sur des statistiques qui comparent des moyennes ou des pourcentages à l'intérieur des classes avec les moyennes ou les pourcentages obtenus sur l'ensemble des données à classer [Lebart et al.,



2006]. En complément de ces statistiques, il existe également des aides graphiques à l'interprétation comme les dendogrammes, les graphiques de densité, etc [Ohsumi, 1988].

Pour sélectionner les variables continues ou les modalités des variables nominales les plus caractéristiques de chaque classe, on mesure l'écart entre les valeurs relatives à la classe et les valeurs dans l'ensemble de données. Ces statistiques peuvent être converties en un critère appelé *valeur-test* qui permet d'opérer un tri sur les variables, et de désigner ainsi les variables les plus caractéristiques [Morineau, 1984].

Nous nous intéressons ici seulement au calcul du valeur-test pour des variables nominales. Une description de ce calcul pour des variables continues est présentée dans [Lebart et al., 2006]. Dans le cas des valeurs-test pour les variables nominales, il s'agit de trouver les modalités dans chaque classe dont l'*abondance* permet de dire qu'elles sont représentatives de la classe. L'abondance de la modalité  $j$  dans la classe  $k$  est mesurée en comparant son pourcentage dans la classe :

$n_{kj} / n_k$  avec son pourcentage dans la population  $n_j/n$

$n_{kj}$  est le nombre de données dans la classe  $k$  avec la modalité  $j$  ;

$n_k$  est le nombre de données dans la classe  $k$ ,

$n_j$  est le nombre de données dans l'ensemble à classer avec la modalité  $j$

$n$  est le nombre total de données dans l'ensemble à classer.

La valeur-test de la modalité  $j$  dans la classe  $k$  est donnée par l'expression :

$$ValeurTest(n_{kj}) = \frac{n_{kj} - E(n_{kj})}{S_k(n_{kj})}$$

Où  $E(n_{kj})$  est l'espérance de  $n_{kj}$  donnée par :

$$E(n_{kj}) = n_k \times \frac{n_j}{n}$$

et  $S_k^2(n_{kj})$  est la variance de  $n_{kj}$  donnée par :

$$S_k^2(n_{kj}) = n_k \times \frac{n - n_k}{n - 1} \times \frac{n_j}{n} \times \left(1 - \frac{n_j}{n}\right)$$

Ce calcul de la valeur-test centre la variable  $n_{kj}$  autour de son espérance est suppose qu'elle est une variable aléatoire qui suit une distribution hypergéométrique. Les modalités représentatives d'une classe sont celles qui donnent des valeurs fortes de la valeur absolue des valeurs-test associées. À partir du calcul des valeurs-test, il est possible de définir un marquage des classes avec les modalités les plus représentatives. La méthode la plus simple est de choisir les  $m$  modalités avec les valeurs-tests les plus significatives. Il existe aussi des

techniques plus perfectionnées comme le marquage sémantique [Morineau et *al.*, 1995] qui prend en compte des combinaisons de différentes modalités pour calculer les valeurs-test. Ces combinaisons sont exprimées sous forme de propositions logiques connectées par les opérateurs ET et OU. Dans ce type de technique, on tient compte des valeurs-tests des modalités isolées, mais aussi des valeurs-tests de toutes les propositions logiques possibles. Par exemple, si l'on considère les caractéristiques *genre (femme ; homme)* et *état civil (marié ; célibataire)*, le marquage sémantique prend en compte les valeurs tests de chaque modalité isolée (*femme, homme, marié, célibataire*) mais aussi de toutes les propositions logiques possibles (*femme ET marié ; femme ET célibataire ; femme OU homme ET marié ; etc.*).

Il existe un certain nombre d'autres méthodes d'interprétation automatique d'une partition. Par exemple, des méthodes basées sur des formalismes de représentation symbolique telles que la méthode CABRO [Bao, 1988 ; 1991] ou la méthode ADS [Diday, 1991 ; 1992]. Une présentation détaillée en est faite dans [Youness, 2004].

### **3.2. Modèles multi-agents de formation d'agents composés**

Les méthodes de classification automatique ont pour objectif de trouver des classes de données semblables. Elles donnent une description synthétique sous forme d'une partition d'un ensemble de données souvent complexe (données décrites par un nombre important de variables différentes avec un nombre important de modalités ou de valeurs par variable). On retrouve ainsi des macro-structures (les classes) qui représentent l'ensemble de données. Dans cette partie, nous présentons des techniques de formation de macro-structures à partir de principes différents. Il s'agit souvent dans ces techniques-là de considérer des micro-entités identiques (des gouttes d'eau, des particules dans un fluide, etc.) qui se rassemblent pour former des macro-entités. Par la suite, les macro-entités évoluent au gré de leurs interactions. Cette approche nous intéresse particulièrement car elle propose un moyen de « rendre vivantes » les macro-structures. On définit, en effet, une méthode de réification automatique des macro-structures et de définition automatique des règles d'interaction et d'évolution des macro-structures. Les modèles présentés sont des modèles multi-agents qui proposent la réification automatique d'agents composés à partir de l'agrégation d'agents plus simples. Ces

modèles ont la particularité de représenter des problèmes physiques, biologiques ou encore robotiques dont les règles d'agrégation et d'interaction entre agents composés sont très bien définies. Ces règles découlent de lois physiques ou biologiques plus ou moins bien maîtrisées. Deux de ces travaux portent sur la modélisation des dynamiques des fluides. Le premier de ces travaux (Bertelle et *al.*, 2002 ; Tranouez et *al.*, 2003) porte sur la modélisation de structures de vortex dans un fluide. Dans ce travail, l'auteur considère le fluide comme un ensemble de particules élémentaires caractérisées par leur position et leur vitesse. Il combine un double système de formation de structures de vortex et d'interaction entre structures. Ainsi, les structures de vortex sont formées par un algorithme d'agglomération de particules voisines dont les vecteurs de vitesse sont similaires. Une fois les vortex formés, ils interagissent entre eux en se renforçant ou en s'annihilant mutuellement. En effet deux règles sont définies pour des vortex qui rentrent en collision : si les deux vortex tournent dans la même direction, leur stabilité est renforcée, si les deux vortex tournent dans des directions opposées, leur stabilité est amoindrie. Lorsqu'un vortex est renforcé, il s'agrandit en assimilant un certain nombre de particules qui sont dans son voisinage. Au contraire, quand la stabilité d'un vortex est amoindrie, sa taille diminue par la perte d'un certain nombre de particules dans sa périphérie. Ce système permet de réaliser des simulations très réalistes des comportements de l'évolution des vortex dans un fluide.

Le deuxième travail est le modèle RIVAGE [Servat, 2000], qui porte sur la simulation de la formation de macro-structures hydrologiques à partir du rassemblement de gouttes d'eau. Dans ce travail, plusieurs types de macro-structures sont considérées (mares d'eau, cours d'eau, ravines, etc.). Les gouttes d'eau se déplacent sur un terrain (modélisé par un automate cellulaire) suite principalement aux effets de la topographie. Les gouttes d'eau se rassemblent dans les dépressions topographiques. Selon la forme de la dépression et la quantité de gouttes d'eau, les rassemblements se transforment en agents « mares », « cours d'eau », « ravines », etc. Une fois la macro-structure formée, les règles d'interaction et déplacement sont décidées par les lois de la dynamique de fluides.

Un troisième exemple de formation automatique d'agents composés est le travail de L. Bonté [Bonté, 2005]. Dans ce travail, l'auteur présente l'exemple de la modélisation de la reproduction de *dictyostelium discoideum* (une espèce d'amibe unicellulaire). Dans certaines conditions (manque de nourriture, présence d'une source lumineuse, etc.), ces amibes se rassemblent en formant des amas globulaires. En présence d'une source de lumière, chaque amas se transforme successivement en ver et en champignon. Une fois l'état de champignon atteint, les amibes localisées dans la tête du champignon se reproduisent par un mécanisme de

dispersion des spores. Suite à la dispersion des spores, les champignons se désintègrent et les amibes se dispersent. Dans ce phénomène, les mécanismes généraux qui gouvernent les transitions entre les différents états (amas, ver, champignon, désintégration) sont maîtrisés (en particulier la transition entre amas et une structure avec un comportement autonome).

Finalement, nous présentons un exemple qui porte sur la robotique, il s'agit du projet MAAM (Molecule = Atome | Atome + Molecule) [Duhaut, 2002 ; Malenfant & Denier, 2003 ; Dubois et *al.*, 2003 ; Brener et *al.* 2004]. L'objectif de ce projet est de définir, spécifier, concevoir et développer un robot modulaire (molécule) constitué d'atomes homogènes capables de s'auto-assembler selon différentes morphologies en fonction d'une tâche à accomplir. Chaque atome est une structure mécanique à six pattes et est capable de se solidariser avec d'autres en s'interconnectant par l'intermédiaire des pattes. Une plateforme de simulation et de visualisation en 3D a été développée, et quelques prototypes réels ont été construits. La tâche la plus basique est l'assemblage entre deux atomes : alignement de pattes et accrochage. D'autres tâches ont également été considérées : le déplacement d'une molécule et le déplacement des atomes ou molécules vers un attracteur, par exemple.

### **3.3. Conclusion sur la formation et l'interaction des groupes**

Nous avons présenté plusieurs algorithmes de classification automatique de données. Dans les algorithmes dits classiques, nous avons constaté que les partitions qu'ils produisent sont dépendantes de plusieurs facteurs, notamment : le traitement dans l'étape d'initialisation, la mesure de distance utilisée, et dans certains cas du paramétrage propre à l'algorithme. La plupart de ces algorithmes, à l'exception des algorithmes hiérarchiques ne sont pas déterministes, la reproductibilité des résultats est donc compromise. Toutefois, ces algorithmes ont été utilisés dans un grand nombre de problèmes et ont démontré leur pertinence à trouver une description adéquate des ensembles divers de données. D'un autre côté, les algorithmes basés sur des agents proposent une vision alternative au problème de la classification. Dans ces algorithmes, on fait appel aux principes de localité et de non centralité propres aux systèmes multi-agents pour produire des regroupements de données similaires dans un espace à deux ou trois dimensions. Une des caractéristiques les plus favorables de ces algorithmes est la visualisation du processus et des résultats de la classification. Ces algorithmes produisent des tas (algorithmes de tri de fourmis) ou des regroupements

dynamiques (algorithmes basés sur les nuages d'individus) de données similaires, qui sont très faciles à repérer visuellement. Dans ces algorithmes, la détection finale des classes est normalement faite par un algorithme classique de classification automatique qui repère les regroupements de données et construit des classes. Ces algorithmes présentent toutefois le même problème de reproductibilité des résultats, car très souvent ils font appel à des calculs aléatoires comme la détermination de la position initiale des données et/ou des fourmis. De plus, ils sont aussi dépendants de leur propre paramétrage.

Dans la deuxième partie de ce chapitre, nous avons exposé quelques exemples qui traitent du problème de construction automatiquement des macro-structures capables d'agir dans le système par interactions entre elles et avec leurs composants élémentaires. D'une part, les composants élémentaires considérés sont des éléments identiques (des gouttes d'eau, des particules dans un fluide, des bactéries, des éléments mécaniques standards). D'autre part, les règles et les conditions de formation des macro-structures reposent sur des règles physiques ou biologiques bien maîtrisées. Ces macro-structures représentent des objets physiques réels.

Dans notre cas d'étude, il s'agit de la formation automatique de méso-agents qui représentent des groupes sociaux et des types d'habitats. Ces méso-agents doivent se former de manière automatique et doivent interagir entre eux et avec les micro-agents (ménages et logements) qui les constituent. Les groupes sociaux et les types d'habitats ne sont pas des objets physiques réels, mais des abstractions qui ont pour objet de donner une vision synthétique de la réalité. En raison de leur nature abstraite, les règles de formation des groupes sociaux et des types d'habitats sont beaucoup moins évidentes que pour les cas d'objets réels. Tout dépend de ce qui est représenté par ces abstractions.

Dans notre cas, nous considérons un groupe social comme un ensemble de ménages semblables quant à la description socio-économique du chef de ménage et un type d'habitat comme un ensemble de logements semblables quant à la description de la qualité du bâti. Dans ces définitions, nous parlons de ménages ou de logements *semblables* quant à leur description. Il y a donc une mesure de similarité, mesurée sur un ensemble de variables. La description socio-économique du chef de ménage et la description de la qualité du bâti d'un logement sont en effet des descriptions complexes composés par un ensemble de variables. A partir de ces constats, il nous semble que la meilleure manière de traiter le problème de la formation des groupes sociaux et des types d'habitats est d'utiliser un algorithme de classification automatique. Le choix de l'algorithme n'est pas facile, nous en proposons plusieurs et les évaluons par rapport à la qualité des partitions produites et par rapport à la qualité des résultats des simulations. À la fin de la partie II de ce document, nous présentons

les différents algorithmes de classification proposés. Dans la partie III, nous présentons une évaluation des différents algorithmes de classification proposés.

Il reste à résoudre la question des interactions entre les agents. Il faut se rappeler que le but de ces interactions est de reproduire les mobilités résidentielles intra-urbaines au niveau mésoscopique. Encore une fois, la difficulté pour définir ces interactions réside dans le caractère abstrait des groupes sociaux et des types d'habitats. Pour pouvoir les définir, nous partons de l'hypothèse selon laquelle le mécanisme de déménagement dans une ville dépend d'un système de préférences localisées et des coûts des déménagements entre secteurs urbains [Piron et al, 2007]. Ainsi, nous proposons un mécanisme fondé sur la concurrence entre groupes sociaux pour acquérir des types d'habitats. Le mécanisme proposé est un mécanisme basé sur des enchères de types d'habitats. Ce mécanisme est présenté au début de la partie II de ce document. Ce mécanisme est aussi abstrait que les méso-agents et sa validation ne peut se faire que par l'évaluation de sa capacité à reproduire la distribution de la population réelle au niveau de la ville.

## **Chapitre IV : L'évaluation des simulations multi-agents**

L'évaluation est une phase très importante du processus de conception et de développement de n'importe quel modèle. Il s'agit, d'une part, de valider le modèle, c'est-à-dire d'évaluer sa capacité à reproduire un comportement réel ; et d'autre part, de le vérifier, c'est-à-dire de garantir qu'il n'y a pas d'erreurs d'implémentation [Schlesinger et *al.*, 1979]. Nous discutons dans ce chapitre principalement des techniques de la validation du modèle.

Il existe différentes méthodes de validation, beaucoup d'entre elles sont présentées dans (Sargent 2005, 2001, 1985). Normalement, il s'agit de mesurer des variables dans le phénomène réel et de les comparer avec les mêmes variables mesurées dans les résultats de la simulation (les variables à mesurer dépendent évidemment du problème qui est modélisé). D'autre part, une série de tests sont nécessaires pour démontrer la pertinence et la précision du modèle pour représenter la situation réelle. Les tests les plus communs sont la validité interne et l'analyse de sensibilité. La validité interne est nécessaire quand il s'agit de modèles stochastiques. C'est une analyse de la variabilité des résultats sur un nombre important d'exécutions avec les mêmes paramètres d'entrée : une variabilité élevée peut mettre en question la validité du modèle (Turner et *al.*, 2001 ; Parker et *al.*, 2003; Sargent, 2001). L'analyse de sensibilité consiste à étudier l'effet du changement dans les données d'entrée et des paramètres internes dans le comportement et les résultats du modèle. Les effets de ces changements devraient être les mêmes dans les résultats du modèle et dans le phénomène réel. Les paramètres qui sont sensibles (ceux qui produisent des changements considérables dans les résultats du modèle) doivent être déterminés de manière très précise avant l'utilisation du modèle.

Pour pouvoir définir les différentes mesures de validation, la première étape est de définir clairement l'objectif du modèle. Dans le cas des modèles spatialisés, quand les prévisions précises sont le but principal, des mesures de l'exactitude des résultats spatiaux sont nécessaires. Quand le but est de représenter un processus et d'expliquer les modèles généraux qui sont observés à travers une variété de situations, il pourrait être nécessaire d'évaluer à quel point le modèle reproduit les propriétés critiques du système en termes de dynamique spatiale et temporelle (Brown et *al.*, 2005; Rand et *al.*, 2003). Dans tous les cas, la validation

doit comporter des jugements sur la capacité du modèle à atteindre les objectifs définis par le modélisateur (Casti, 1997 ; Brown et *al.* 2004), ce qui dépend du choix entre les aspects réels que l'on inclut dans le modèle et ceux que l'on décide d'ignorer.

La validation est traditionnellement réalisée en comparant les résultats du modèle soit avec des données ou observations du monde réel ; soit avec les résultats d'un autre modèle (Parker et *al.*, 2003). Cette comparaison est habituellement effectuée en utilisant des méthodes statistiques qui permettent d'établir une corrélation raisonnable entre les sorties du modèle et les vraies données. La validation semble être une question critique pour n'importe quelle approche de modélisation appliquée à n'importe quel système, mais dans le cas des SMA utilisés dans la modélisation des systèmes complexes la validation peut s'avérer particulièrement difficile. Un certain nombre d'auteurs soulignent ces difficultés, qui sont récapitulées ci-dessous :

- La nature des systèmes complexes : Les caractéristiques qui définissent un système complexe, (les rétroactions, la capacité d'adaptation, etc.) rendent ces systèmes très incertains et donc, très difficiles (ou impossibles) à prévoir. Ces caractéristiques rendent difficile la validation des modèles basés sur des SMA, en particulier quand le niveau de modélisation microscopique est très fin (Brown et *al.*, 2004 ; Rand et *al.*, 2003).
- La disponibilité des données : Parfois on ne dispose pas des données nécessaires à la validation du modèle. Cette situation est très courante dans le cas de l'étude des dynamiques urbaines, où étant donnée la taille des problèmes étudiés l'acquisition des données peut être coûteuse et difficile (Sargent, 2005).
- Le processus de validation : Dans les modèles basés sur des SMA, il est souvent nécessaire de valider le comportement du modèle ou, en d'autres termes, de valider les différentes dynamiques qui conduisent au résultat final analysé. L'étude de chaque dynamique sous une forme désagrégée est l'un des points primordiaux dans l'évaluation des SMA. Idéalement, le comportement dynamique des agents devrait être comparé à de vraies données. Mais les données réelles au niveau individuel, avec le détail exigé pour la comparaison, sont rarement disponibles.

En plus de ces questions, lorsqu'il s'agit de modèles géographiques, nous nous confrontons souvent à un problème très bien connu, qui est la dépendance des résultats de la validation à l'échelle de mesure et du niveau d'analyse observation. (Lam & Quattrochi 1992 ; Saura & Martínez-Millán 2001) D'après l'expérience, on observe toujours de meilleurs ajustements entre les résultats simulés et la situation réelle lorsque l'on compare les résultats à petite



échelle, avec un faible niveau de détail, que lorsque l'on compare ces résultats à des échelles plus fines. De même, les résultats s'ajustent mieux lorsqu'on les valide au niveau macroscopique ou à des niveaux intermédiaires, que lorsqu'on les valide au niveau microscopique.

Dans notre cas, nous proposons une stratégie de validation basée sur la comparaison de la distribution de la population réelle et la distribution de la population simulée à l'année finale de simulation. Nous proposons le calcul d'un indice qui permet de vérifier l'exactitude de la distribution de la population obtenue à partir des simulations. Ce mécanisme est présenté à la fin de la partie II. D'autre part, nous réalisons une analyse de sensibilité aux paramétrages du modèle proposé, ainsi qu'une analyse par rapport à l'échelle de validation considérée. Les résultats de ces analyses sont présentés dans la partie III de ce document.

## **Partie II : Modèles de simulation multi-agent**

## **Chapitre V : Introduction au modèle**

### **5.1. Présentation générale**

L'objectif principal du modèle que nous avons développé est de reproduire l'évolution de distribution spatiale de la population d'une ville sur une période de temps donné. Pour atteindre notre objectif, nous adoptons une vision synthétique de la ville et nous proposons un modèle de simulation multi-agent avec les caractéristiques suivantes : nous considérons comme unités principales de modélisation des groupes de ménages (groupes sociaux) et des groupes de logements (types d'habitats). Ces groupes sont des typologies obtenues à partir de la population réelle de ménages et du parc de logements par un processus de classification automatique. Dans la suite du document, lorsque nous parlons de mécanisme de formation de groupes, nous faisons référence à la classification des données qui décrivent la population de ménages suivie de la constitution de groupes sociaux à partir de la partition obtenue, et aussi à la classification des données qui décrivent le parc de logements suivie de la constitution des types d'habitats à partir de la partition obtenue. Etant donné le caractère abstrait des groupes considérés et les données multi-variées qui décrivent la population des ménages et le parc de logements, tel que nous l'avons exposé dans la première partie de ce document, nous considérons que la classification automatique est le mécanisme le plus adapté pour constituer les groupes.

La relation entre groupes sociaux et types d'habitats (quel groupe social occupe quels types d'habitats dans tel ou tel lieu) est exprimée sous la forme d'une matrice de distribution de la population. Dans un premier temps, nous proposons un modèle dit *synthétique* où l'on ne considère qu'un seul niveau de modélisation : le niveau mésoscopique. On y considère les groupes sociaux et les types d'habitats ainsi que leurs relations. Ensuite, nous étendons ce modèle pour travailler à un niveau microscopique où l'on considère des ménages et des logements et leurs relations (quel ménage occupe quel logement). Nous appelons ce deuxième modèle le *modèle étendu*. La proposition de ces deux modèles nous permet de tester la nécessité de considérer en même temps les niveaux mésoscopique et microscopique pour reproduire l'évolution de la distribution de la population au lieu de considérer uniquement le

niveau mésoscopique. Ce test est fait par la comparaison et l'analyse des résultats de différentes simulations en utilisant les deux modèles. Ces analyses sont le sujet de la partie III de ce document.

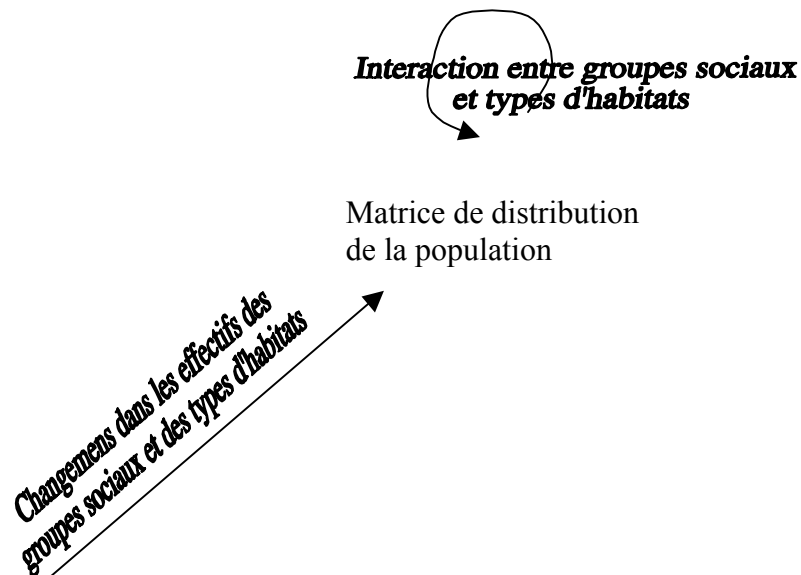


Figure 9. Modèle de simulation synthétique: uniquement le niveau mésoscopique.

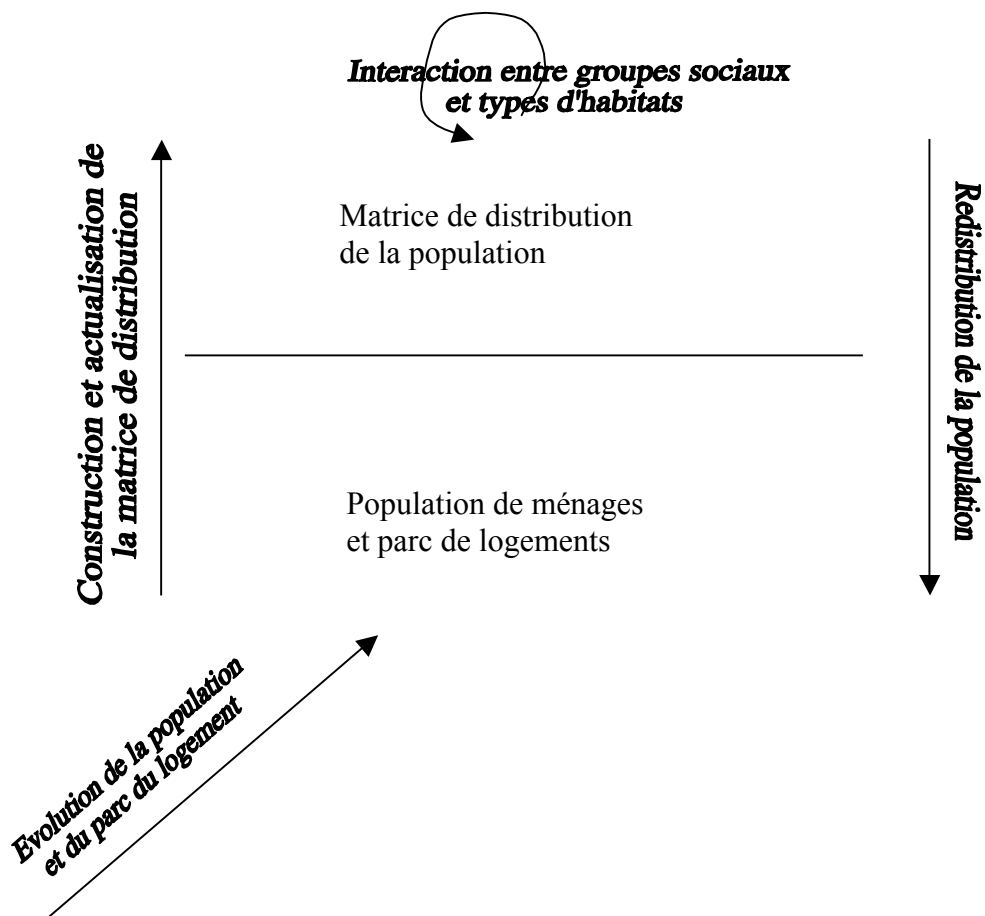
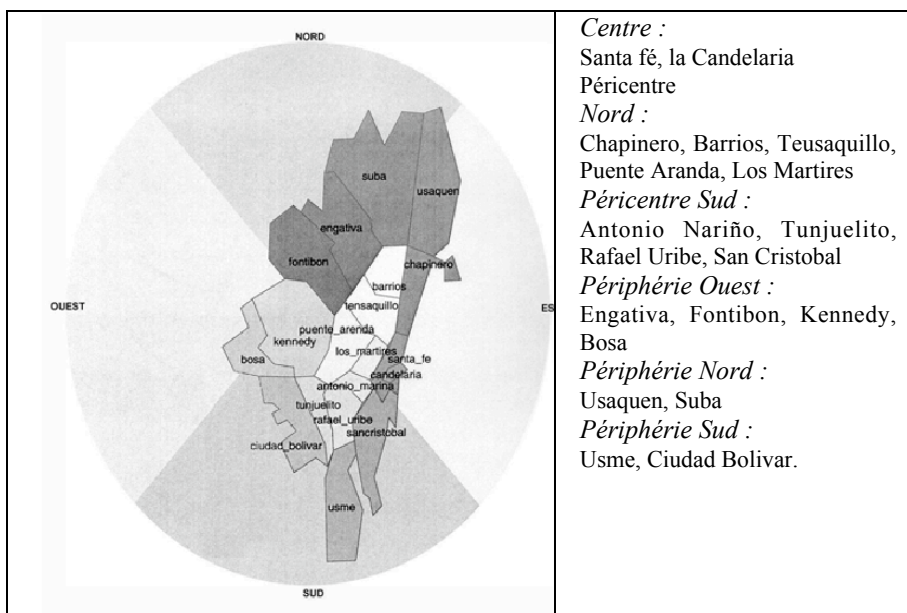


Figure 10. Modèle de simulation étendu : Interaction entre les niveaux microscopique et mésoscopique.

Les deux modèles développés sont présentés de manière schématique dans les figures ci-dessus. Deux composantes principales sont considérées comme moteur de l'évolution de la distribution de la population de la ville : un mécanisme de mobilité fondé sur la concurrence entre groupes sociaux pour obtenir des logements libres et un mécanisme d'évolution de la population. Nous modélisons la concurrence entre groupes sociaux par un mécanisme basé sur la vente aux enchères des logements libres. Dans le cas du modèle synthétique, les conséquences de la concurrence entre groupes sont des changements dans les relations entre groupes sociaux et types d'habitats. Dans le cas du modèle étendu, on considère aussi les changements des relations ménages-logements.

Le mécanisme d'évolution est différent dans les deux modèles proposés : dans le modèle synthétique, le mécanisme d'évolution agit directement au niveau mésoscopique; dans le modèle étendu, le mécanisme agit d'abord au niveau microscopique et les effets sont transmis au niveau mésoscopique. Les résultats des mécanismes d'évolution sont observés au niveau macroscopique (ensemble de secteurs urbains ou arrondissements, voir carte de Bogotá ci-dessous).



Carte 1. les 19 arrondissements et les 6 zones urbaines de Bogotá

Nous avons développé une plateforme de simulation multi-agent en langage C++ dans laquelle les deux modèles sont implémentés. Les différents composants considérés dans cette plateforme sont présentés dans le diagramme ci-dessous. Dans les chapitres qui suivent, nous présentons chacun de ces composants. Nous commençons par présenter quelques définitions fondamentales (5.2.). Nous continuons par la présentation du mécanisme de mobilité (package mobilité) dans le chapitre VI, et ensuite, dans le chapitre VII nous présentons les modèles

synthétique et étendu (packages « Distribution de la population », « Evolution » et en partie « Distribution Microscopique »). Dans le chapitre VIII nous présentons le mécanisme de formation de groupes (packages « Formation de Groupes » et « Distribution Microscopique »). Nous finissons cette partie du document par la présentation du mécanisme de validation des résultats (chapitre IX).

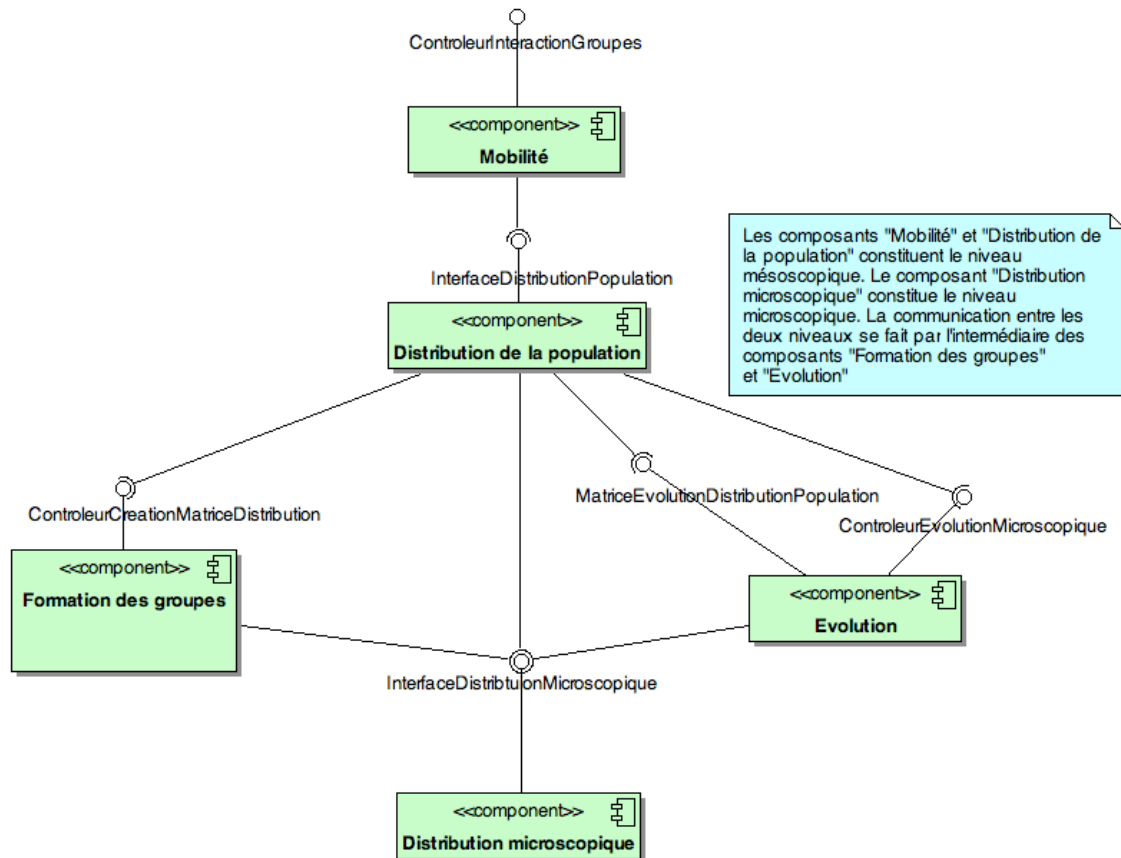


Figure 11. Composants de la plateforme de simulation multi-agent développée

Le schéma général de simulation des deux modèles est présenté dans l’encadré ci-dessous sous forme d’algorithme. Ce schéma va nous servir de guide dans la présentation de chaque mécanisme et modèle tout au long de cette partie du document. Dans cet algorithme, les étapes essentielles de la simulation multi-agent sont définies. Cet algorithme reçoit comme entrée les populations réelles initiale et finale pour la période à simuler et le nombre d’années de la période. Les paramètres additionnels dépendent du modèle de distribution de la population utilisé et définissent notamment la manière d’initialiser les paramètres d’évolution et de faire évoluer la population simulée.

Simulation (*population initiale réelle, population finale réelle, nombre d’années, paramètres additionnels*)

DEBUT

1. Initialiser les paramètres d’évolution de la population

```

2. Initialiser la population simulée à partir de la population initiale réelle.
3. POUR année=1 jusqu'au nombre d'années
    FAIRE
        a. Evoluer la population simulée.
        b. Actualiser la répartition de la population simulée
        c. année _ année + 1
    FIN POUR
4. Valider la population simulée contre la population finale réelle
FIN

```

Encadré 3. Macro algorithme principal de simulation

## 5.2. Définitions

Nous présentons ci-après un ensemble de définitions nécessaires pour la description des différents mécanismes dans les chapitres suivants.

- **Micro-entités considérées** : Ménages et logements.
- **Profil** : Description des caractéristiques d'une micro-entité ou des caractéristiques « moyennes » d'un groupe de micro-entités. C'est un vecteur binaire représentant un ensemble de caractéristiques sous forme disjonctive complète (cf. Annexe II, Représentation des profils). Dans l'implémentation, un profil est représenté comme un *long long*, ce qui permet de faire des opérations bit à bit entre paires de profils. On peut ainsi, par exemple, définir de manière simple et efficace la distance de *Hamming* (cf. Annexe III, Calcul de distances entre profils) entre deux profils. Dans l'annexe I, on présente les caractéristiques et modalités retenues pour décrire les ménages et logements de la ville de Bogota.
- **Groupe** : Représente un ensemble de micro-entités. Il est défini par un profil « moyen » et le nombre de micro-entités qu'il représente. Il existe deux types de groupes : les groupes sociaux formés par des ménages et les types d'habitats formés par des logements.
- **Distribution microscopique de population** : Elle contient la distribution des micro-entités dans la ville. Elle est composée de trois listes :
  - a. Ménages avec logements (ou logements occupés) : c'est une liste de triplés profil-ménage/profil-logement/secteur-urbain.

- b. Ménages sans logement : liste de profil-ménage.
- c. Logements libres : liste de couples profil-logement/secteur-urbain.

▪ **Matrice d'occupation** : Elle représente la distribution socio-spatiale au niveau des groupes d'un secteur urbain. C'est un tableau croisé de types d'habitats avec des groupes sociaux. Chaque ligne représente les effectifs d'un certain type d'habitat et chaque colonne représente les effectifs d'un certain groupe social. La dernière colonne de la matrice contient les effectifs de logements libres par type d'habitat. À titre d'exemple, si l'on suppose qu'il existe 5 types d'habitats et 6 groupes sociaux (voir tableaux ci-dessous), un exemple de matrices d'occupation pour les secteurs urbains hypothétiques SU1 et SU2 est présenté dans les tableaux ci-dessous.

Type d'habitat	TH1	TH2	TH3	TH4	TH5
Effectifs	43497	591145	131081	199158	310972

Tab. 1 : Exemple de liste de types d'habitat : les effectifs représentent le nombre total de logements qui composent chaque habitat dans la ville

Groupe social	GS1	GS2	GS3	GS4	GS5	GS6
Effectifs	109156	143225	274892	165295	267079	316206

Tab. 2 : Exemple de liste de groupes sociaux : les effectifs représentent le nombre total de ménages qui composent chaque groupe social dans la ville

	GS1	GS2	GS3	GS4	GS5	GS6	Log. Libres	Totaux
TH1	86	122	186	199	296	145	68	1102
TH2	205	153	306	271	440	70	86	1531
TH3	229	443	655	431	852	414	69	3093
TH4	257	303	505	453	631	498	160	2807
TH5	155	201	275	192	259	415	896	2393
Totaux	932	1222	1927	1546	2478	1542	1279	10926

Tab. 3 : Exemple de matrice d'occupation du secteur urbain « SU1 »

	GS1	GS2	GS3	GS4	GS5	GS6	Log. Libres	Totaux
TH1	537	52	1461	21	2565	367	212	5215
TH2	204	51	1225	674	469	9	33	2665
TH3	328	155	438	1083	15	34	0	2053
TH4	159	137	12	46	0	2979	11	3344
TH5	0	1062	84	225	10	24	0	1405
Totaux	1228	1457	3220	2049	3059	3413	256	14682

Tab. 4 : Exemple de matrice d'occupation du secteur urbain « SU2 »



- **Matrice de distribution de la population** : c'est une synthèse de la distribution réelle de la population. Elle représente la distribution socio-spatiale au niveau des groupes d'un ensemble de secteurs urbains. Elle est composée d'une liste de groupes sociaux, d'une liste de types d'habitats, de plusieurs matrices d'occupation, et d'une liste d'effectifs de ménages (classés par groupe social) sans logement (voir exemple dans le tableau ci-dessous). Chaque matrice d'occupation représente la distribution de la population dans un secteur urbain. Les groupes sociaux et les types d'habitats sont déterminés par un processus de classification automatique de la distribution réelle de la population.

Groupe social	GS1	GS2	GS3	GS4	GS5	GS6
Ménages sans logement	4333	3879	2470	5405	3902	4428

Tab. 5 : Exemple de liste de ménages sans logement: les effectifs représentent le nombre total de ménages sans logement par groupe social dans la ville

- **Coût de déménager** : le coût de déménager représente la perte socio-économique liée à un déménagement entre deux secteurs urbains. Il dépend de plusieurs facteurs, par exemple : la distance euclidienne entre secteurs urbains, la perte en termes d'infrastructure. La distance euclidienne peut, par exemple, servir de mesure de la « perte sociale » produite par l'éloignement du secteur urbain où sont installés la famille et les amis d'un ménage. La perte en termes d'infrastructures peut représenter une perte en termes d'accès aux services et transport : par exemple, déménager d'un secteur urbain avec une bonne prestation de services publics vers un secteur urbain avec une mauvaise prestation implique un coût de déménager important.

Le coût de déménager est symétrique quand il n'est fonction que de la distance entre secteurs urbains. C'est-à-dire qu'il est le même quand on déménage du secteur A vers le secteur B que quand on le fait dans le sens inverse. Quand on considère d'autres facteurs, par exemple la perte en termes d'infrastructures, le coût de déménager n'est pas forcément symétrique : lorsqu'on déménage d'un secteur urbain riche en prestation de services urbains vers un secteur pauvre, il y a une perte en termes d'accès aux services, mais lorsqu'on déménage d'un secteur urbain pauvre vers un riche, il y a au contraire un gain. Dans le premier cas, on parle d'un coût positif, dans le second d'un coût négatif (pour représenter le gain). Dans le cas général, le coût de déménager dépend de plusieurs facteurs. Si, par exemple, on considère en même temps les facteurs « distance » et « accès aux services », le coût n'est pas symétrique. La définition du coût est dépendante du problème que l'on veut modéliser.

Nous considérons des coûts de déménager normalisés entre 0 et 1. L'utilisation d'une telle échelle définit les coût maximum et minimum et permet le traitement simplifié des coûts. S'il existe des gains lors du déménagement entre deux secteurs, il suffit de définir un seuil entre perte et gain dans l'échelle normalisée. Ainsi, par exemple, on peut considérer que tout coût inférieur à 0,5 est un gain, 0 représentant le gain maximal, et les coûts supérieurs à 0,5 représentent des pertes lors du déménagement.

- **Graphe d'interconnexion** : C'est un graphe orienté dans lequel les nœuds sont des secteurs urbains et le poids d'un arc représente le coût de déménager du secteur urbain origine de l'arc vers le secteur urbain destination de l'arc. Étant donné que le coût de déménager entre deux secteurs urbains n'est pas forcément symétrique, il faut définir des arcs dans les deux sens.

Nous considérons que des graphes d'interconnexion sont complets quand tous les nœuds sont connectés avec tous les autres nœuds par un arc directement. Les coûts de déménager non définis sont considérés comme maximum : si un arc entre deux secteurs n'a pas été défini, un arc par défaut est créé avec un coût de déménagement de 1.

		Destination				
		SU1	SU2	SU3	SU4	SU5
Origine	SU1	0	0,87	0,95	0,92	0,22
	SU2	0,82	0	0,13	0,84	0,36
	SU3	0,16	0,36	0	0,66	0,27
	SU4	0,64	0,84	0,12	0	0,28
	SU5	0,29	0,43	0,55	0,19	0

Tab. 6 : Exemple de graphe d'interconnexion sous forme matricielle

Dans cet exemple, les lignes représentent les secteurs urbains d'origine et les colonnes les secteurs urbains de destination. Par exemple, le coût de déménager du secteur urbain SU3 vers le secteur urbain SU4 est de 0,66 (ligne SU3, colonne SU4). Les 0's dans la diagonale veulent dire que le coût de déménager d'un secteur urbain vers lui-même est minimal, ce n'est pas forcément toujours le cas, par exemple lorsque l'on considère uniquement les différences d'accès aux services entre secteurs urbains, il est moins coûteux de déménager vers les secteurs urbains mieux desservis que de rester dans un secteur urbain d'origine mal desservi.

## Chapitre VI : Les interactions entre groupes sociaux dans le marché du logement, un mécanisme de mobilité

La première composante que nous considérons dans l'évolution de la distribution spatiale de la population d'une ville est la concurrence entre groupes sociaux pour acquérir des logements libres. Nous supposons que cette concurrence dépend principalement de deux facteurs : un système localisé de préférences et la demande de logements.

Le système localisé de préférences exprime l'hypothèse que les groupes sociaux préfèrent certains types d'habitats sur d'autres et que leurs préférences changent selon le secteur urbain. Les préférences sont déterminées à partir des matrices d'occupation.

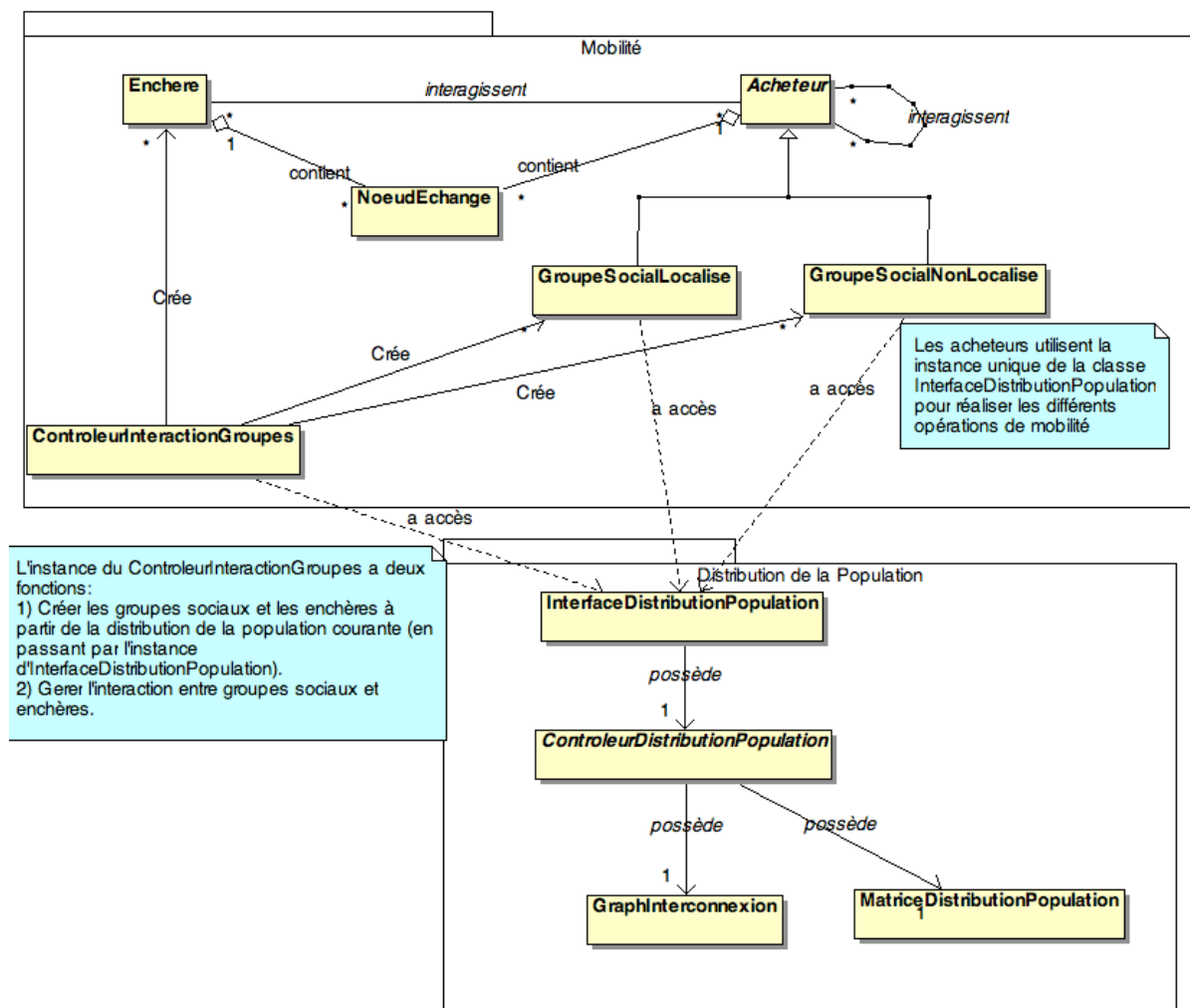


Figure 12. Diagramme de classe. Niveau mésoscopique.

Pour représenter la concurrence entre groupes sociaux, nous avons implémenté un système multi-agent (*package* « Mobilité » dans le diagramme ci-dessus), où les groupes sociaux (acheteurs) interagissent à travers un mécanisme de vente aux enchères qui permet l'échange de logements vides.

## 6.1. Calcul de la préférence localisée

La préférence d'un groupe social pour un type d'habitat dépend des matrices d'occupation. La préférence est en effet localisée, c'est-à-dire que la préférence dépend du secteur urbain. La préférence d'un groupe social pour un type d'habitat dans un secteur urbain donné est égale au pourcentage d'occupation du type d'habitat par le groupe social dans le secteur urbain donné. Par exemple, si l'on considère la matrice d'occupation du secteur urbain SU1 (cf. définition de matrice d'occupation dans le chapitre précédent), les préférences des différents groupes sociaux par les différents types d'habitats dans le secteur SU1 sont présentées dans le tableau ci-dessous.

	GS1	GS2	GS3	GS4	GS5	GS6
TH1	0,092	0,1	0,097	0,129	0,119	0,094
TH2	0,22	0,125	0,159	0,175	0,178	0,045
TH3	0,246	0,363	0,34	0,279	0,344	0,268
TH4	0,276	0,248	0,262	0,293	0,255	0,323
TH5	0,166	0,164	0,143	0,124	0,105	0,269
TOTAUX	1	1	1	1	1	1

Tab. 7 : Exemple du calcul des préférences dans le secteur urbain « SU1 »

Chaque valeur de préférence est le résultat de la division du nombre de logements occupés par un groupe social dans un type d'habitat donné (pris de la matrice d'occupation) par le nombre total d'effectifs du groupe social dans le secteur urbain (somme de la colonne correspondant au groupe social dans la matrice d'occupation). Par exemple, la préférence du groupe social GS1 par le type d'habitats TH1 dans le secteur urbain SU1 est égale à  $86 / 932 = 0,092$ , où 86 est le nombre de logements du type d'habitats TH1 occupés par GS1 dans le secteur urbain SU1 et 932 est le nombre total de logements occupés par GS1 dans le secteur urbain SU1, c'est-à-dire la somme de la colonne GS1 dans la matrice d'occupation de SU1.

Les préférences prennent des valeurs entre 0 et 1. La somme des valeurs des préférences d'un groupe social dans un secteur urbain donné est égale à 1.

## 6.2. Les agents du système

Nous considérons deux types d'agents : les enchères et les acheteurs. Les acheteurs représentent des groupes sociaux qui souhaitent déménager vers certains types d'habitats proposés par les enchères.

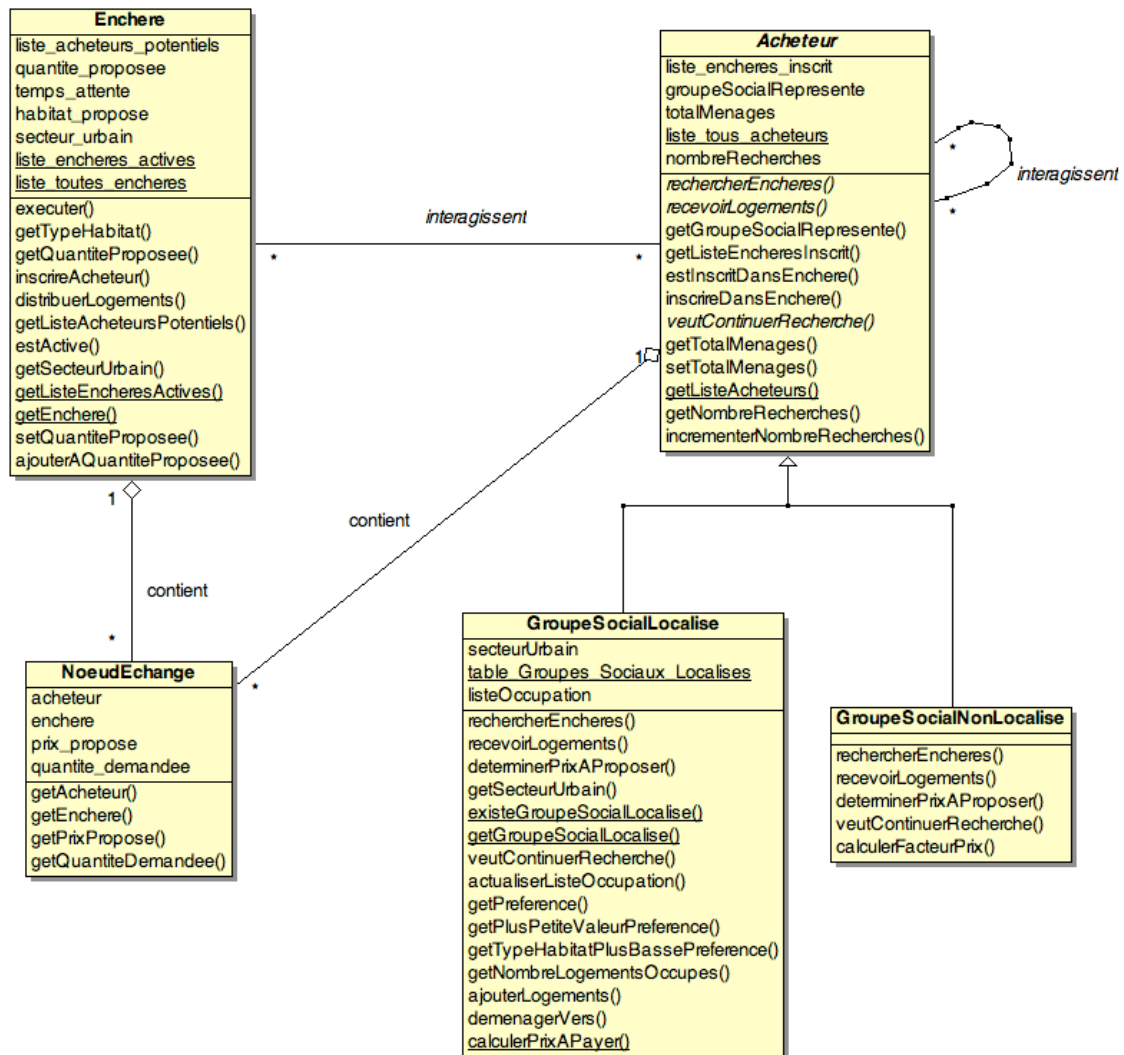


Figure 13. Diagramme de classe. Agents du système

### 6.2.1. L'agent enchère

Nous considérons des enchères multi-objet [Houssein et *al.*, 2001], c'est-à-dire des enchères où plusieurs logements vides d'un même type d'habitat sont proposés et où les acheteurs proposent un prix unitaire et une quantité à acheter. Ainsi, une enchère (voir diagramme ci-dessous) est définie essentiellement par le profil du type d'habitat proposé (*habitat\_propose*),

une quantité de logements proposés (*quantite\_proposee*) et le secteur urbain où ces logements sont proposés (*secteur\_urbain*). L'enchère possède une liste de nœuds d'échange, c'est-à-dire des objets qui contiennent une référence à un acheteur, une référence à l'enchère, la quantité désirée et le prix unitaire proposé par l'acheteur. Cette liste est appelée liste d'acheteurs potentiels. Elle est ordonnée selon le prix unitaire proposé par chaque acheteur.

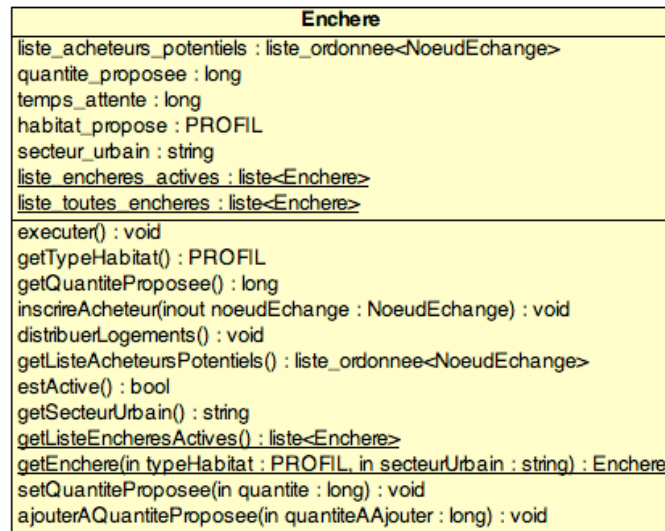


Figure 14. Diagramme de classe. La classe *Enchere*

Dans la vente aux enchères, nous ne considérons qu'un seul tour, c'est-à-dire que les acheteurs potentiels ne proposent qu'une seule fois un prix unitaire et une quantité à acheter. Ces prix et quantités n'évoluent pas au cours d'une même vente aux enchères. L'exécution de la vente aux enchères se fait en deux étapes principales :

- Inscription des acheteurs : Des nouveaux acheteurs s'inscrivent à l'enchère en proposant un prix unitaire et une quantité désirée de logements.
- Distribution des logements : L'enchère distribue les logements parmi les acheteurs inscrits selon le prix unitaire proposé et la quantité désirée.

L'exécution de la vente aux enchères dépend de trois états possibles :

- Enchère active/inactive : Une enchère n'est active que si le nombre de logements qu'elle propose est supérieur à 0. Les enchères actives sont contenues dans la liste d'enchères actives.
- Enchère en attente d'acheteurs.
- Enchère en fermeture : l'enchère n'accepte plus d'inscription de nouveaux acheteurs et passe à l'étape de distribution des logements.

L'inscription des acheteurs est seulement permise si l'enchère est active. Une enchère reste un certain temps en attente d'inscription de nouveaux acheteurs. Le changement vers l'état de

fermeture dépend d'une probabilité de changement qui est fonction du temps passé dans l'état d'attente d'acheteurs (voir équation ci-dessous). Plus le temps passé dans cet état est grand, plus la probabilité de passer à l'état de fermeture est grande.

$$probabilitéChangementEtat(tempsAttente) = 1 - e^{\frac{-1 \times tempsAttente}{2}}$$

Le *tempsAttente* est une variable entière qui est initialisée à 0, sa valeur augmente de 1 à chaque fois que l'on exécute l'enchère (voir ci-après).

### 6.2.1.1. Exécution d'une enchère

La méthode *executer* dans la classe *Enchere* contrôle le changement d'état de l'enchère. Si le passage vers l'état de fermeture est déclenché, l'enchère distribue les logements parmi les acheteurs inscrits sur la liste d'acheteurs potentiels (voir distribution de logements). Si, après la distribution, il ne reste plus de logements à distribuer l'enchère passe à l'état d'inactivité (elle est supprimée de la liste d'enchères actives), autrement elle revient à l'état d'attente d'acheteurs.

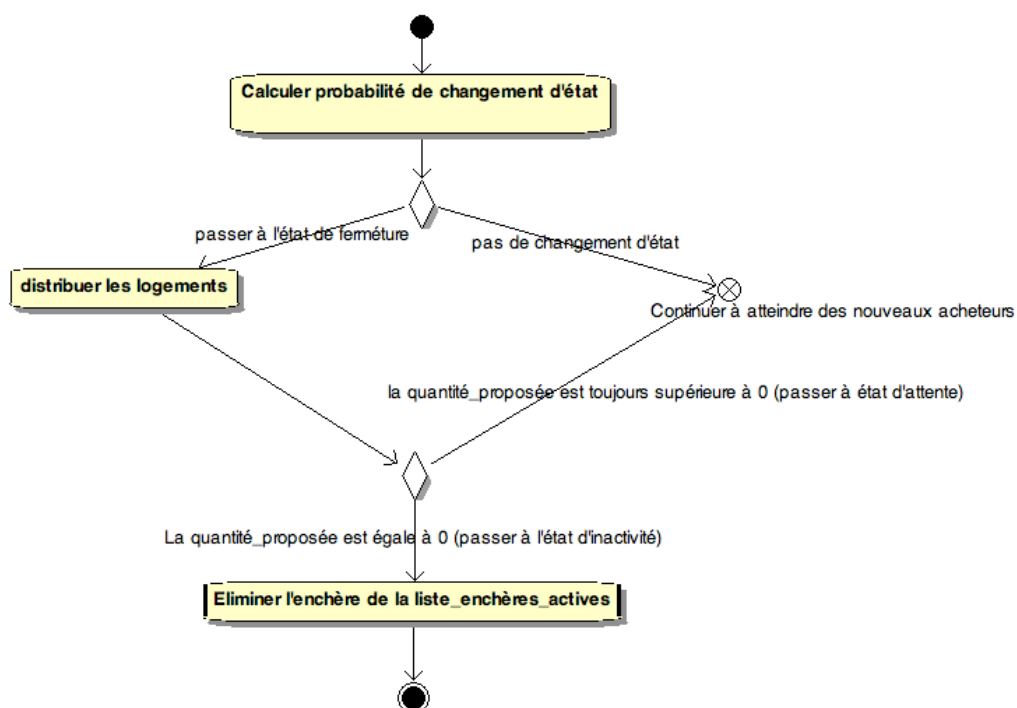


Figure 15. Diagramme de changement d'état d'une enchère (méthode *executer* de la classe *Enchere*)

### 6.2.1.2. Distribution des logements

La distribution des logements proposés par une enchère est faite parmi les acheteurs inscrits sur la liste d'acheteurs potentiels. Les logements sont distribués suivant un ordre descendant selon le prix unitaire proposé : les acheteurs qui ont proposé les prix les plus élevés sont servis

en premier. Les acheteurs sont servis dans la limite de la quantité disponible de logements libres. La quantité réelle distribuée à chaque acheteur dépend du stock et de la quantité acceptée par l'acheteur : d'une part, on ne peut pas donner une quantité plus grande que la taille courante du stock et d'autre part, un acheteur n'est pas obligé d'accepter la quantité qu'il avait demandée originellement. La quantité réellement acceptée ( $q_{Reçue}$ ) est inférieure ou égale à la quantité initialement proposée (voir algorithmes de réception de logements).

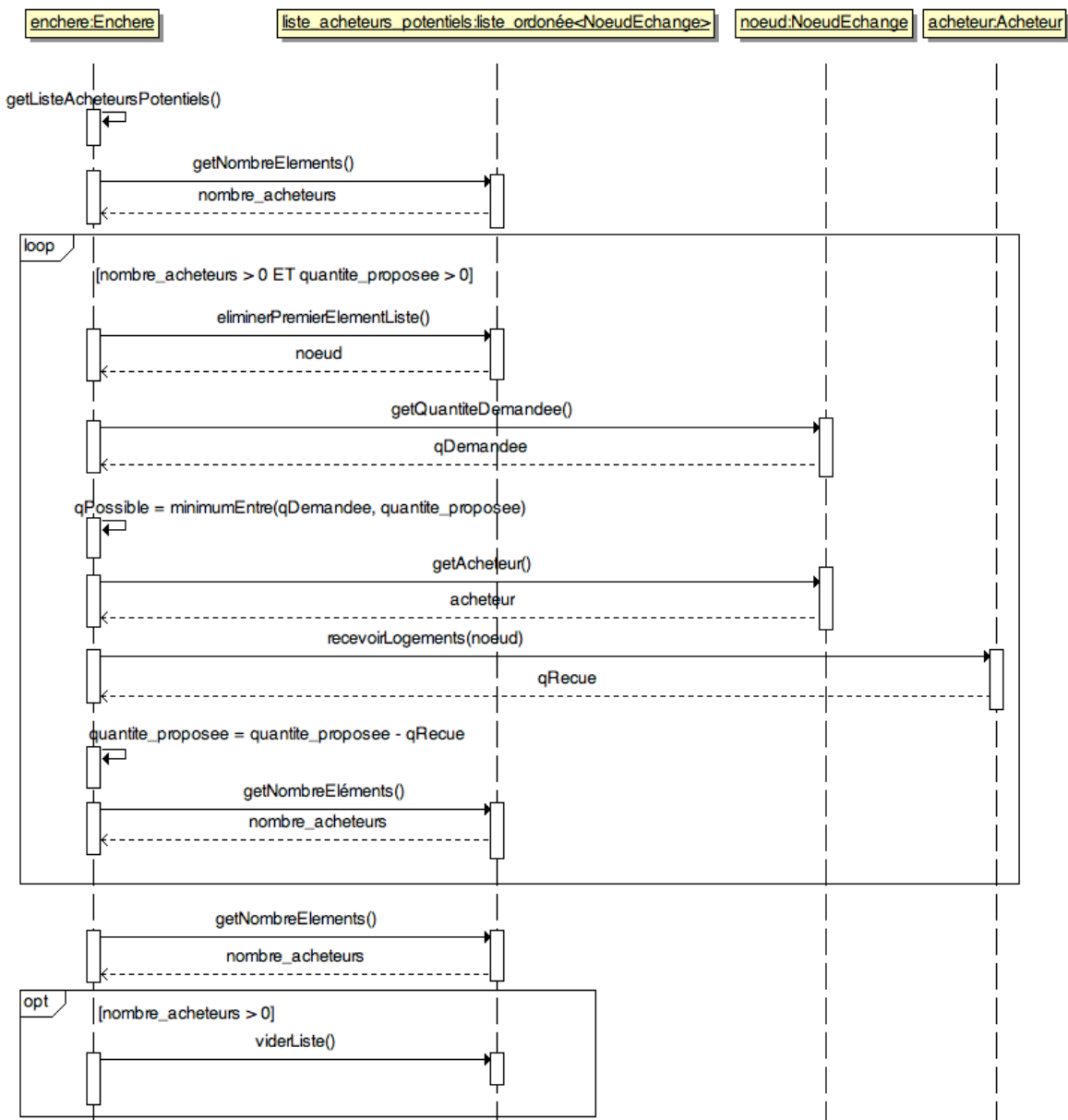


Figure 16. Diagramme de séquence pour la méthode `distribuerLogements()` de la classe `Enchere`



## 6.2.2. Les agents acheteurs

Un acheteur (voir diagramme ci-dessous) a une liste de nœuds d'échange pour chaque enchère dans laquelle il est inscrit (*liste\_encheres\_inscrit*). L'acheteur est identifié par le profil du groupe social qu'il représente (*groupeSocialRepresente*). Un acheteur réalise principalement deux actions :

- La recherche d'enchères.
- La réception de logements.

Ces deux actions sont implémentées différemment selon le type d'acheteur. Il existe deux types d'acheteurs : les groupes sociaux localisés et les groupes sociaux non localisés.

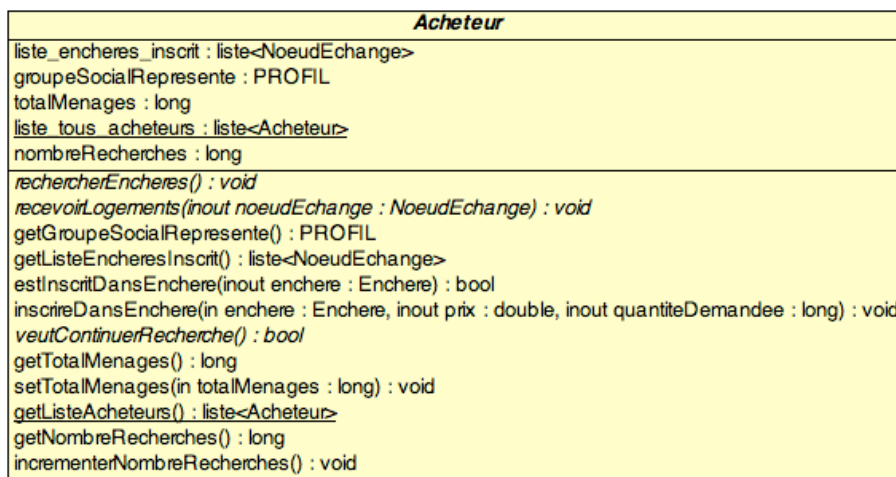


Figure 17. Diagramme de classe. La classe *Acheteur*

### 6.2.2.1. Groupe social localisé

Un groupe social localisé représente un sous-ensemble localisé d'un groupe social. Il représente l'ensemble de ménages qui font parti d'un groupe social et qui occupent des logements localisés dans un secteur urbain en particulier. Autrement dit, un groupe social localisé représente une colonne dans la matrice d'occupation d'un secteur donné. Un groupe social localisé possède une liste d'occupation (de types d'habitat occupés) à partir de laquelle on calcule ses préférences pour des types d'habitat. Cette liste est construite à partir d'une colonne de la matrice d'occupation du secteur urbain occupé par le groupe.

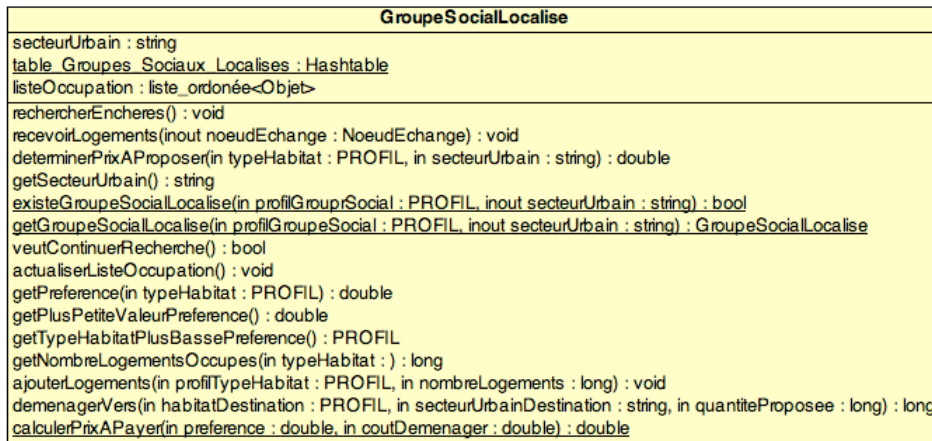


Figure 18. Diagramme de classe. La classe *GroupeSocialLocalise*

Par exemple, le groupe social localisé correspondant au groupe social GS1 dans le secteur urbain SU1 possède la liste d'occupation suivante :

Liste d'occupation du groupe localisé GS1-SU1 : {TH4 : 257 ; TH3 : 229 ; TH2 : 205 ; TH5 : 155; TH1 : 86 }

Cette liste d'occupation est construite à partir de la colonne GS1 de la matrice d'occupation du secteur urbain SU1 (voir la définition de matrice d'occupation dans 5.2.).

Les préférences calculées à partir de la liste d'occupation sont : {TH4 : 0,276 ; TH3 : 0,246 ; TH2 : 0,22 ; TH5 : 0,166 ; TH1 : 0,092}

La liste d'occupation est ordonnée de manière décroissante ce qui permet de différencier facilement les types d'habitat les plus occupés (avec des préférences plus élevées) de ceux qui le sont moins (avec les préférences minimales).

#### 6.2.2.1.1. Recherche d'enchères par un groupe localisé

Un groupe social localisé, que l'on va appeler *gsl*, s'inscrit seulement dans les enchères actives qui proposent des logements dans des secteurs urbains où le groupe social représenté par *gsl* est déjà présent, c'est-à-dire les secteurs urbains où il existe un groupe social localisé avec le même profil (*groupeSocialRepresente*) que *gsl*. La recherche d'enchères par un groupe social localisé est implémentée dans la méthode *rechercherEncheres* qui est présentée ci-dessous sous forme de diagramme de séquence.

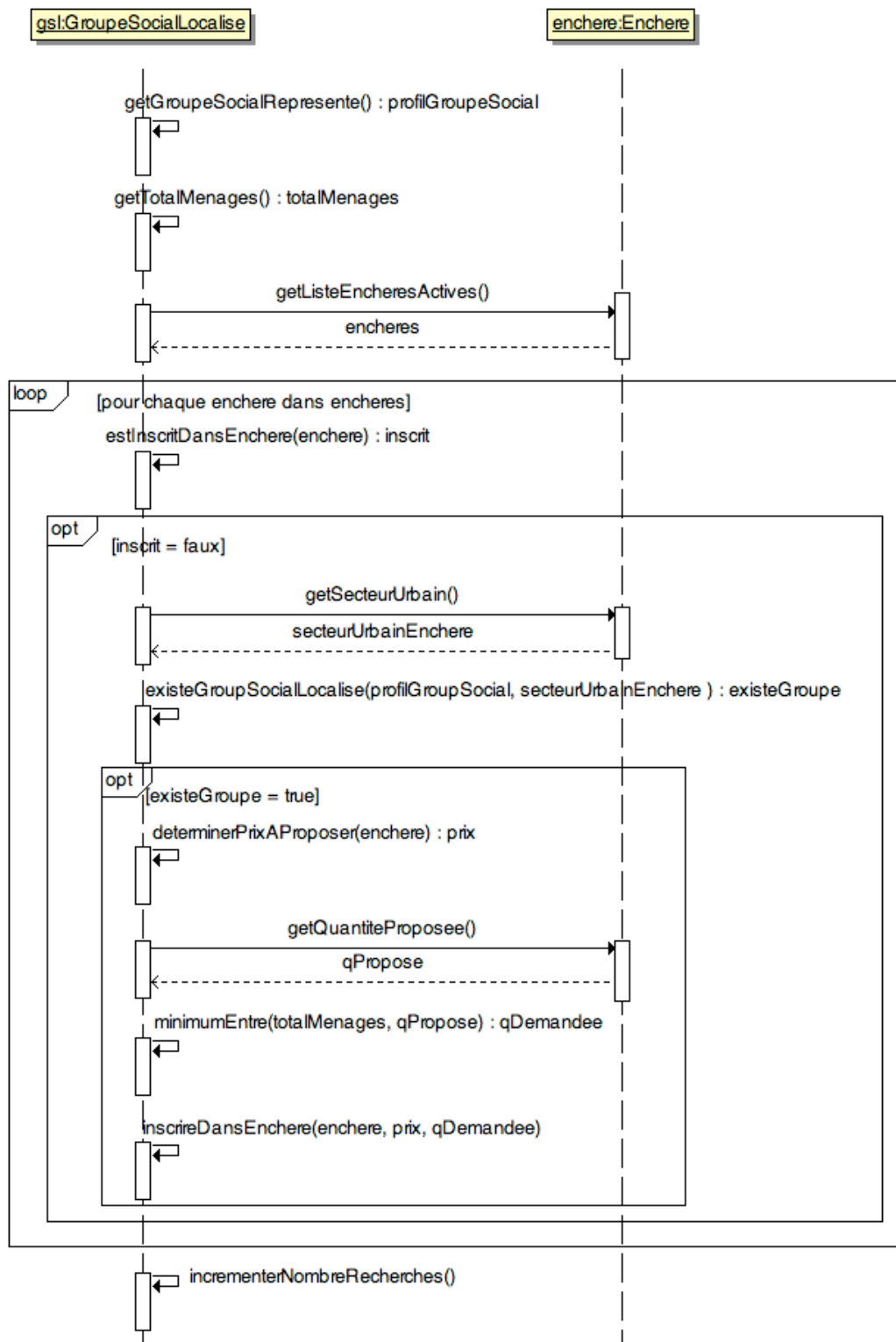


Figure 19. Diagramme de séquence pour la méthode *rechercherEncheres* de la classe *GroupeSocialLocalise*

#### 6.2.2.1.2. Détermination du prix à proposer dans une enchère

Le prix qu'un groupe social localisé *gs/* est prêt à payer dans une enchère dépend de la préférence du groupe social (représenté par *gs/*) pour le type d'habitat dans le secteur urbain de l'enchère et du coût de déménager entre le secteur urbain de *gs/* et le secteur urbain de

l'enchère. L'algorithme de détermination du prix par un groupe social localisé est présenté dans le diagramme ci-après :

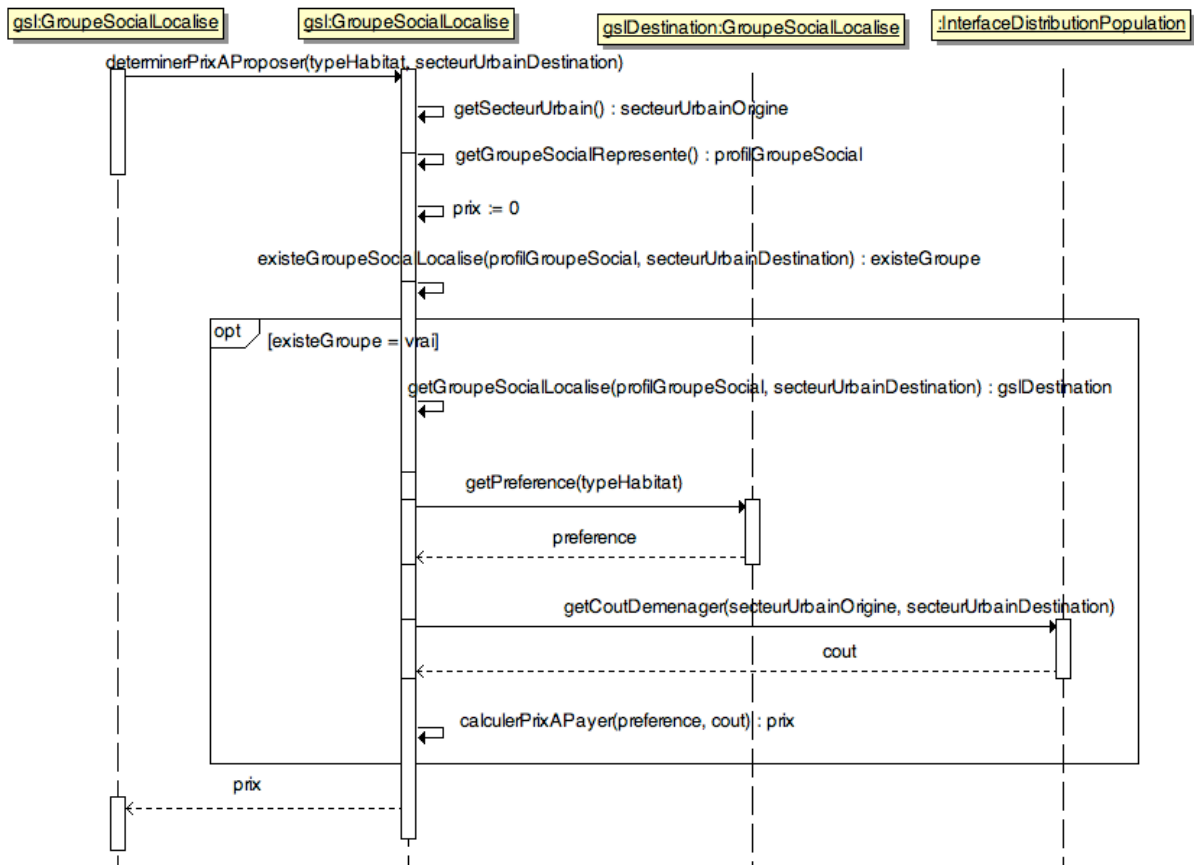


Figure 20 Diagramme de séquence pour la méthode *determinePrixAProposer* de la classe *GroupeSocialLocalise*

La fonction *calculerPrixAPayer* renvoie au prix qu'un groupe social localisé est prêt à payer pour un type d'habitat étant donné la préférence pour ce type d'habitat et le coût de déménager.

Nous imposons trois conditions à cette fonction:

- Elle doit être une fonction décroissante par rapport au coût de déménager.
- Elle doit être une fonction croissante par rapport à la préférence.
- Elle doit être définie pour toutes les valeurs de préférence et de coût de déménager.

Sachant que le coût de déménager et la préférence sont des variables réelles qui prennent des valeurs entre 0 et 1, nous proposons l'expression suivante pour le calcul du prix :

$$\text{calculerPrixAPayer}(\text{coûtDéménager}, \text{préférence}) = \text{préférence} \times (1 - \text{coûtDéménager})$$

Cette expression respecte les trois conditions imposées et renvoie à des prix entre 0 et 1.

### 6.2.2.1.3. Réception des logements



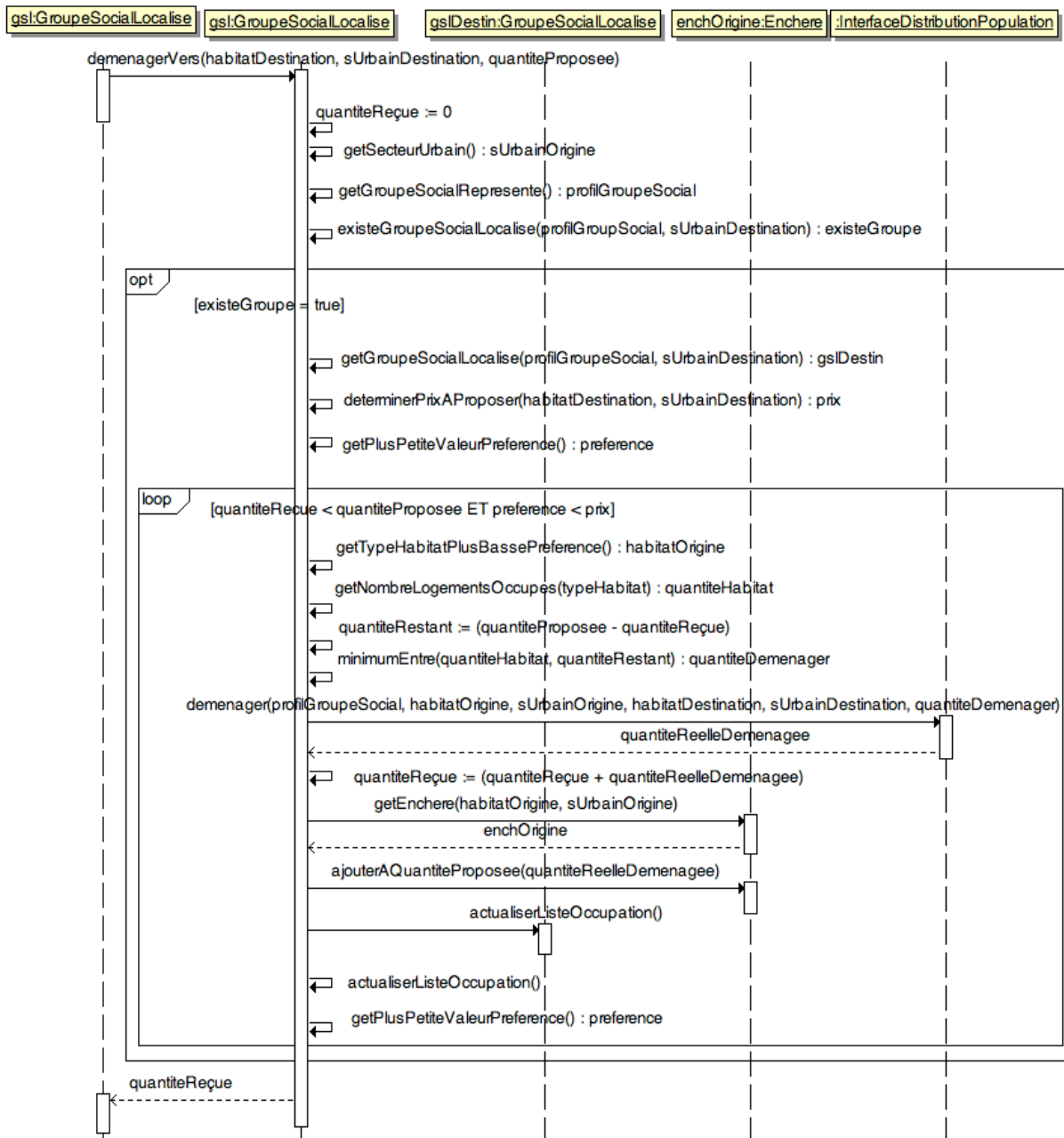


Figure 22. Diagramme de séquence pour la méthode *demenagerVers* de la classe *GroupeSocialLocalise*

### 6.2.2.2. Groupe social non localisé

Un groupe social non localisé représente le sous-ensemble d'un groupe social dont le secteur urbain n'est pas défini, c'est-à-dire les ménages sans logement qui appartiennent au groupe social. Un groupe social non localisé est défini par le nombre de ménages qui le constituent. Pour le cas d'un groupe social non localisé, le nombre de recherches réalisées (hérité de la classe *Acheteur*) représente le temps passé sans trouver de logements par tous ses ménages.

Étant donné qu'un groupe social non localisé n'occupe aucun secteur urbain par définition et qu'il ne possède pas de liste de préférences, les algorithmes de détermination du prix, de recherche et de réception de logements sont différents de ceux d'un groupe social localisé.

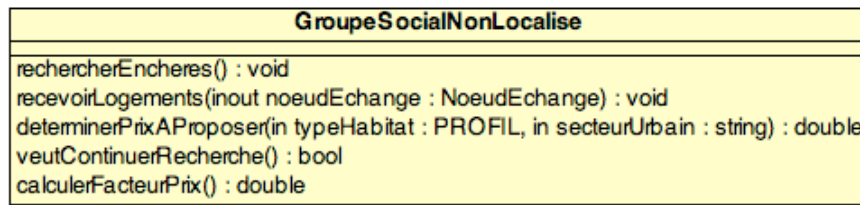


Figure 23. Diagramme de classe. La classe *GroupeSocialNonLocalise*

#### 6.2.2.2.1. Recherche d'enchères

À la différence des groupes localisés, un groupe social non localisé *gsnl* s'inscrit dans toutes les enchères actives. En effet, un *gsnl* s'inscrit même dans des enchères qui sont dans des secteurs urbains où le groupe social représenté par *gsnl* n'est pas présent, c'est-à-dire dans des secteurs urbains où il n'y a pas de groupes sociaux localisés avec le même profil que *gsnl*. Le fait d'accepter toutes les enchères, donne la possibilité de déménager vers des secteurs urbains où le groupe social représenté par *gsnl* n'était pas présent auparavant.

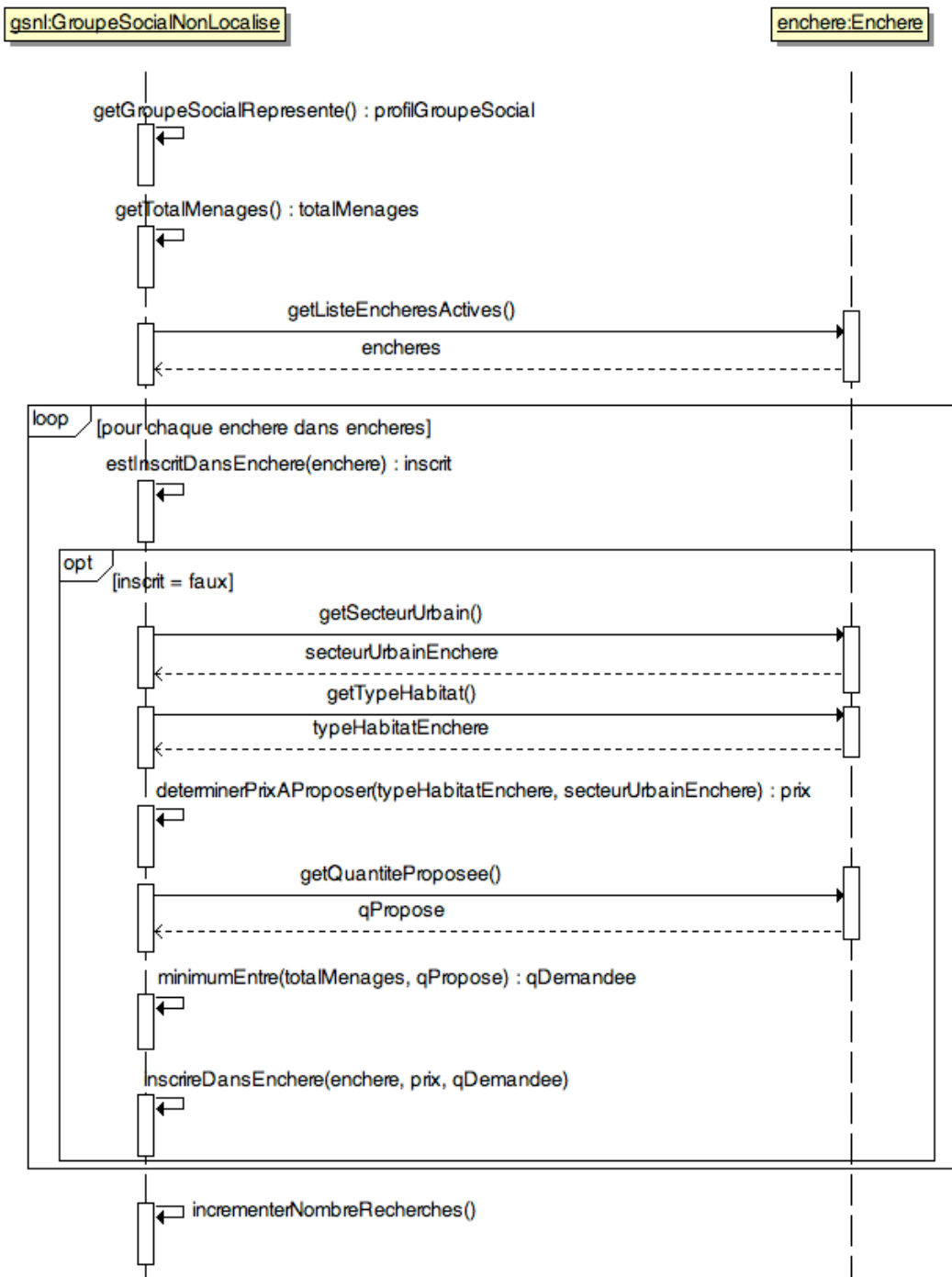


Figure 24. Diagramme de séquence pour la méthode *rechercherEnchères* de la classe *GroupeSocialNonLocalise*

#### 6.2.2.2.2. Détermination du prix à proposer dans une enchère

Dans le cas des groupes sociaux non localisés, le prix à payer pour un certain type d'habitat dépend principalement du temps passé sans avoir trouvé de logements par tous les ménages qui constituent le groupe:



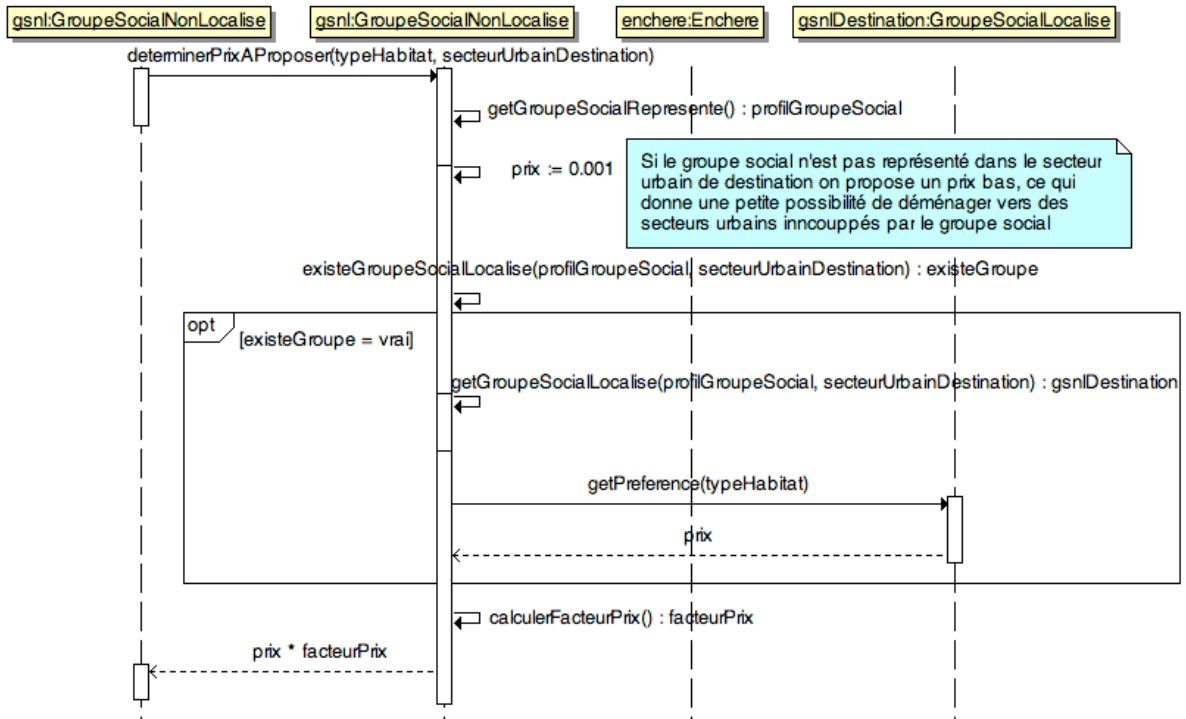


Figure 25. Diagramme de séquence pour la méthode `determinePrixAProposer()` de la classe `GroupeSocialNonLocalise`

Étant donné qu'un groupe social non localisé n'occupe aucun secteur urbain, dans l'algorithme de détermination du prix, on ne peut pas déterminer le coût de déménager à partir du graphe d'interconnexion. Nous considérons donc que le coût de déménager dans ce cas est de 0. D'autre part, dans cet algorithme, on multiplie le prix par un *facteurPrix*, ce facteur est un nombre réel plus grand ou égal à 1 et exprime l'urgence pour résoudre la demande du groupe non localisé. Le *facteurPrix* est donné par la formule suivante:

$$\text{facteurPrix}(\text{nombreRecherches}) = k^{\text{nombreRecherches}}$$

Où :

$k$  est une constante supérieure ou égale à 1. Nous utilisons 1,01 par défaut.

$\text{nombreRecherches}$  est un nombre entier supérieur ou égal à 0, qui est incrémenté à chaque recherche d'enchères. Cette variable exprime le temps passé sans avoir résolu totalement la demande de logements du groupe social non localisé.

Il faut aussi remarquer qu'un groupe social localisé est prêt à payer un prix supérieur à 0 pour un type d'habitat dans un secteur urbain qui n'est pas occupé par le groupe social représenté. De ce fait, on donne une petite probabilité au groupe social de déménager vers des secteurs urbains qu'il n'occupait pas auparavant.

### 6.2.2.2.3. Réception des logements

Un groupe social non localisé déménage vers n'importe quel type de logement dans n'importe quel secteur urbain qu'on lui propose. Il permet ainsi la création de nouveaux groupes localisés dans des secteurs urbains non occupés auparavant.

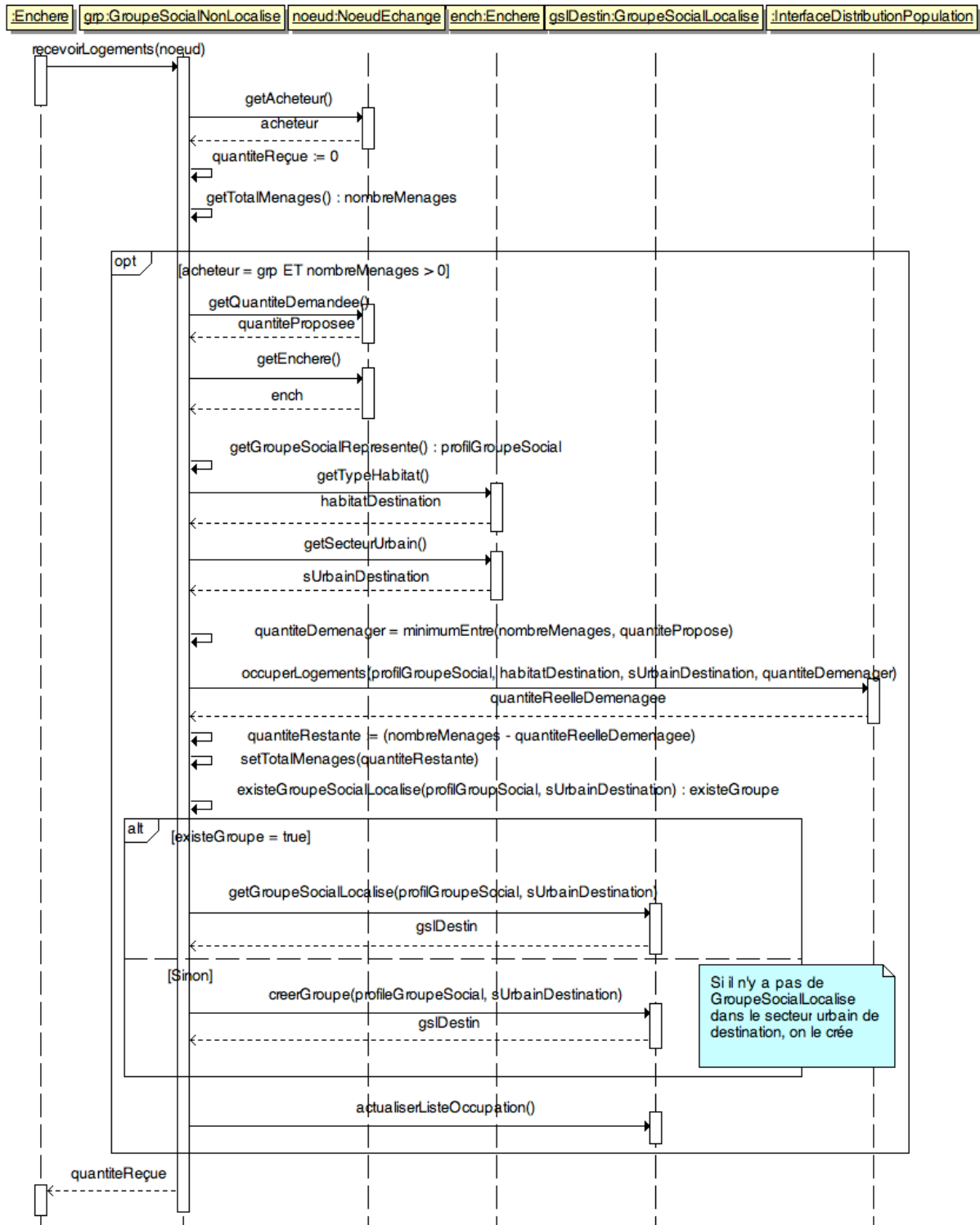


Figure 26. Diagramme de séquence pour la méthode *recevoirLogements()* de la classe *GroupeSocialNonLocalise*

Le nombre de ménages qui déménagent vraiment est plus petit ou égal à la quantité de logements demandés initialement à l'enchère. La valeur de cette quantité dépend de deux facteurs :

- Le nombre courant de ménages dans le groupe social non localisé. Ce nombre a pu diminuer depuis l'inscription à l'enchère.
- Le nombre de ménages qui déménagent vraiment dépend de l'implémentation de la méthode *occuperLogements* dans le contrôleur de la distribution de la population (implémentation de la classe *ControleurDistributionPopulation*).

### 6.3. Le contrôle de l'interaction

Nous avons implémenté un système de simulation pseudo-parallèle similaire à celui utilisé dans la plateforme de simulation multi-agent MMINUS [Vanbergue, 2003]. Dans ce système on ne considère qu'un seul fil d'exécution et un seul agent est exécuté à la fois. Pour éviter des séquences d'exécution prédéfinies, tous les agents exécutables (enchères actives ainsi que groupes sociaux localisés et non localisés) sont mis dans une liste de laquelle ils sont retirés selon un tirage aléatoire. Chaque agent exécute ses actions lorsqu'on le retire de la liste. Cette liste est gérée par un programme informatique que nous appelons le contrôleur de l'interaction entre groupes (classe *ControleurInteractionGroupes*). En plus de la gestion de cette liste, le contrôleur de l'interaction a comme tâche d'initialiser le marché de logement, c'est-à-dire, de créer les enchères initiales et les groupes sociaux initiaux.

L'initialisation du marché du logement et l'exécution des interactions entre groupes sociaux et enchères correspondent à la phase d'actualisation de la répartition de la population, soulignée dans le macro-algorithme principal de simulation ci-dessous :

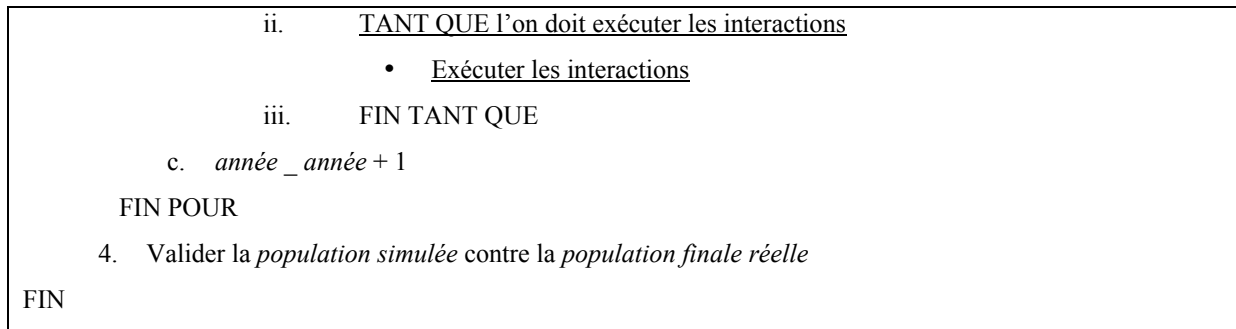
Simulation (*population initiale réelle, population finale réelle, nombre d'années, paramètres additionnels*)

DEBUT

1. Initialiser les paramètres d'évolution de la population
2. Initialiser la *population simulée* à partir de la *population initiale réelle*.
3. POUR *année=1* jusqu'au *nombre d'années*

FAIRE

- a. Evoluer la *population simulée*.
- b. Actualiser la répartition de la *population simulée* :
  - i. Initialiser le marché



Encadré 4. Phase d'actualisation de la répartition de la population simulée dans le macro-algorithme principal de simulation.

### 6.3.1. Initialisation du marché

À fin de créer les groupes sociaux (localisés et non localisés) et les enchères initiales, le contrôleur de l'interaction des groupes accède aux informations sur la distribution courante de la population par l'intermédiaire de l'interface de la distribution de la population (classe *InterfaceDistributionPopulation*).

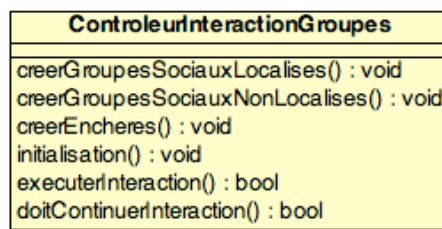


Figure 27. Diagramme de classe. La classe *ControleurInteractionGroupes*

#### 6.3.1.1. Création des groupes sociaux localisés

Les informations nécessaires pour la création de chaque groupe social localisé sont obtenues en interrogeant l'interface de distribution de la population. L'interface de distribution permet au contrôleur de l'interaction de connaître la liste de groupes sociaux, types d'habitat et secteurs urbains. Le contrôleur crée aussi la liste d'occupation de chaque groupe social localisé en interrogeant l'interface de distribution de la population sur le nombre de logements de chaque type d'habitat occupé par le groupe social localisé (message *getNombreLogementsOccupes*).

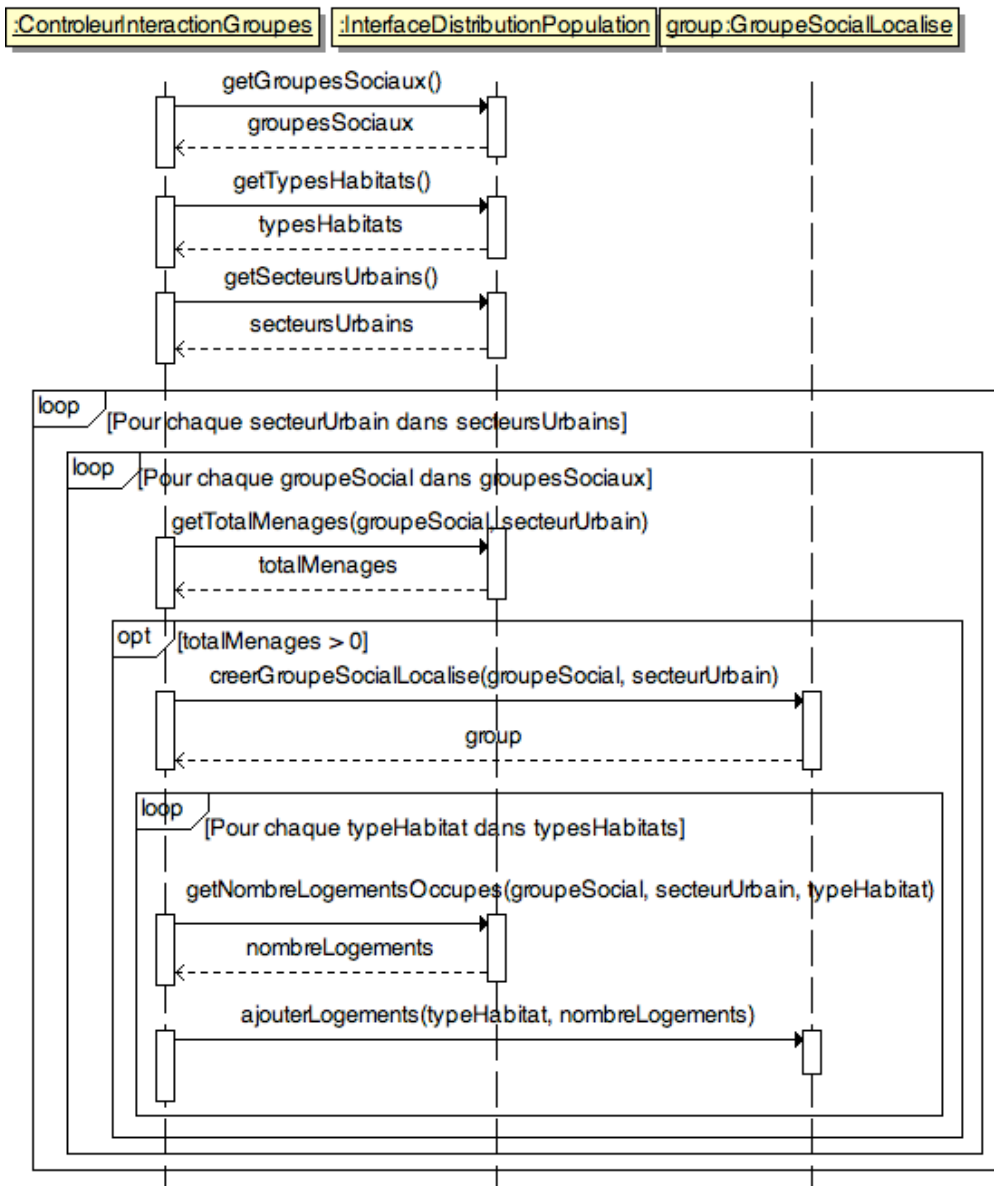


Figure 28. Diagramme de séquence pour la méthode *creerGroupesSociauxLocalises()* de la classe *ControleurInteractionGroupes*

### 6.3.1.2. Création des groupes sociaux non localisés

Le contrôleur de l'interaction crée les groupes sociaux non localisés à partir de la liste de ménages-sans-logement obtenue en interrogeant l'interface de distribution de la population. Il crée un groupe social non localisé par groupe social avec des ménages sans logement.

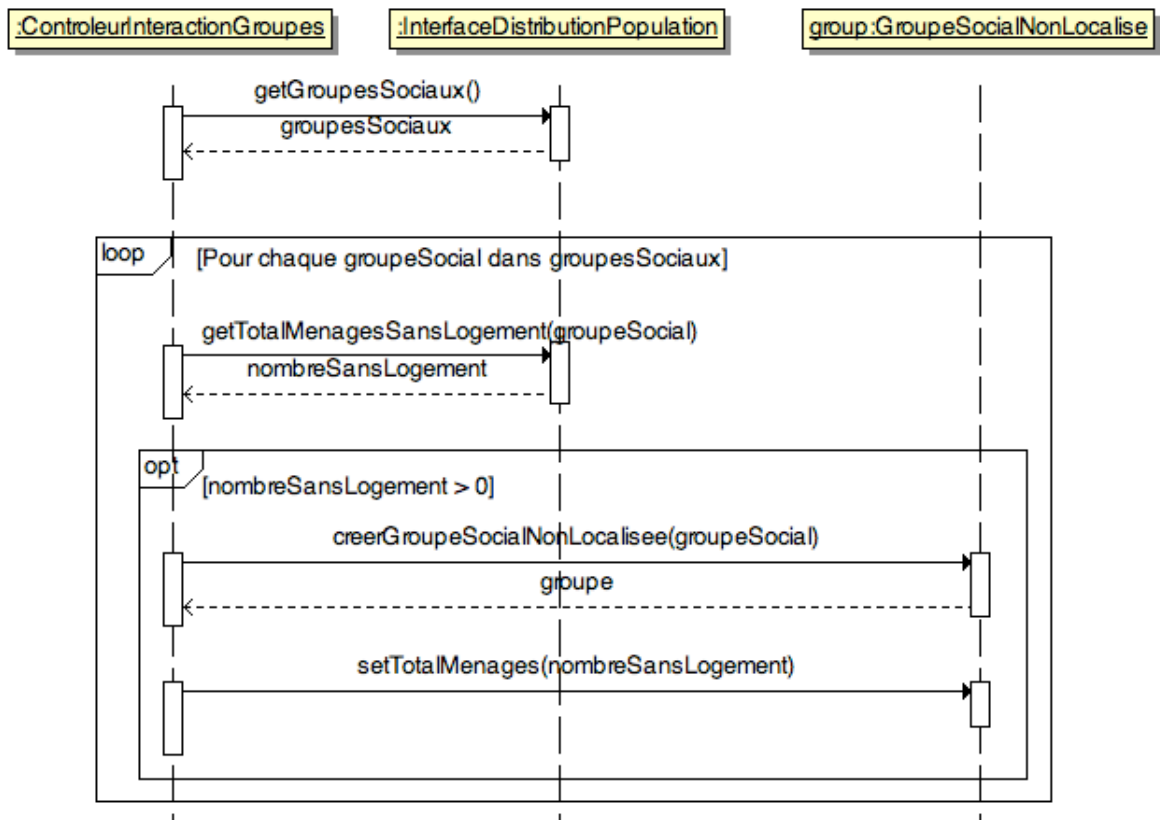


Figure 29. Diagramme de séquence pour la méthode *creerGroupesSociauxNonLocalises()* de la classe *ControleurInteractionGroupes*

### 6.3.1.3. Création des enchères

Le contrôleur de l'interaction crée une enchère par type d'habitat avec des logements libres dans chaque secteur urbain. L'algorithme de création des enchères est présenté dans le diagramme ci-après.

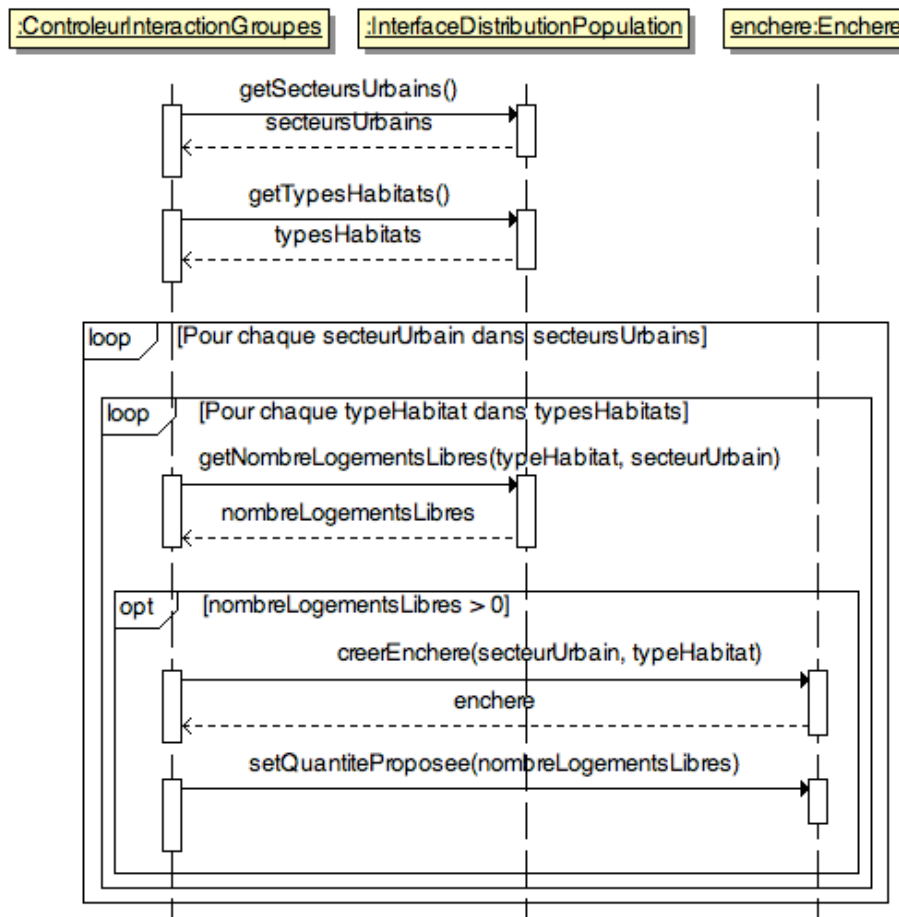


Figure 30 Diagramme de séquence pour la méthode *creerEncheres* de la classe *ControleurInteractionGroupes*

### 6.3.2. Gestion de l'interaction

Les interactions entre les enchères et les groupes sociaux sont contrôlées par les algorithmes d'actualisation implémentés dans le contrôleur de l'interaction. L'algorithme principal d'actualisation est *executerInteraction* (voir diagramme ci-dessous). Dans cet algorithme, tous les agents (enchères et acheteurs) sont mis sur une liste, de laquelle ils sont retirés un par un suivant un ordre aléatoire, pour ensuite être exécutés. Si l'agent retiré de la liste est une enchère, on appelle la méthode *executer* de l'enchère, s'il s'agit d'un acheteur (groupe social localisé ou non localisé), on appelle la méthode *rechercherEncheres* de l'acheteur. L'exécution de l'interaction se répète jusqu'à ce qu'il n'y ait plus d'enchères actives (donc plus de logements libres) ou jusqu'à ce qu'aucun acheteur ne veuille continuer la recherche des nouvelles enchères (cf. diagramme de séquence *doitContinuerInteraction*).

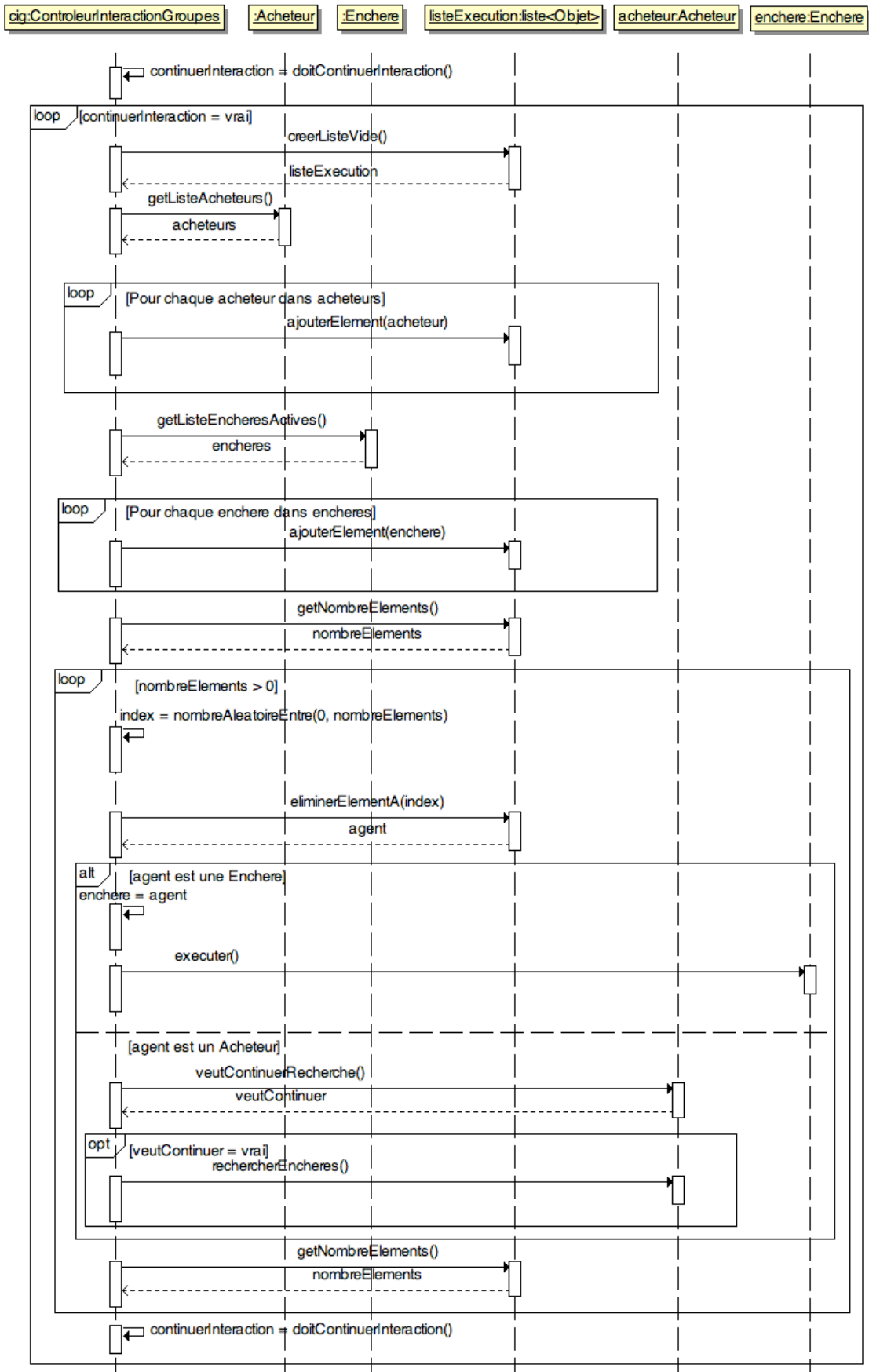


Figure 31. Diagramme de séquence pour la méthode *executerInteraction* de la classe *ControleurInteractionGroupes*



La méthode *doitContinuerInteraction* dans la classe *ControleurInteractionGroupes* définit la condition d'arrêt de l'exécution des interactions entre groupes. Cette méthode renvoie faux (arrêt des interactions) s'il n'existe plus d'enchères actives (il n'y a plus de logements libres) ou si aucun acheteur veut continuer la recherche d'enchères. Autrement, la méthode renvoie *vrai* (les interactions continuent). La décision de continuer la recherche d'enchères de chaque acheteur est implémentée dans la méthode *veutContinuerRecherche* des classes *GroupeSocialLocalise* et *GroupeSocialNonLocalise* :

- L'implémentation dans la classe *GroupeSocialLocalise* tient compte d'une probabilité de continuer la recherche de logements. Cette probabilité dépend du nombre de recherches déjà effectuées (*nombreRecherches* hérité de la classe *Acheteur*). Plus le nombre de recherches est grand plus la probabilité de continuer la recherche est petite.
- L'implémentation dans la classe *GroupeSocialNonLocalise* renvoie *vrai* si le groupe social non localisé a encore des ménages sans logement (*totalMenages > 0*), autrement elle renvoie *faux*.

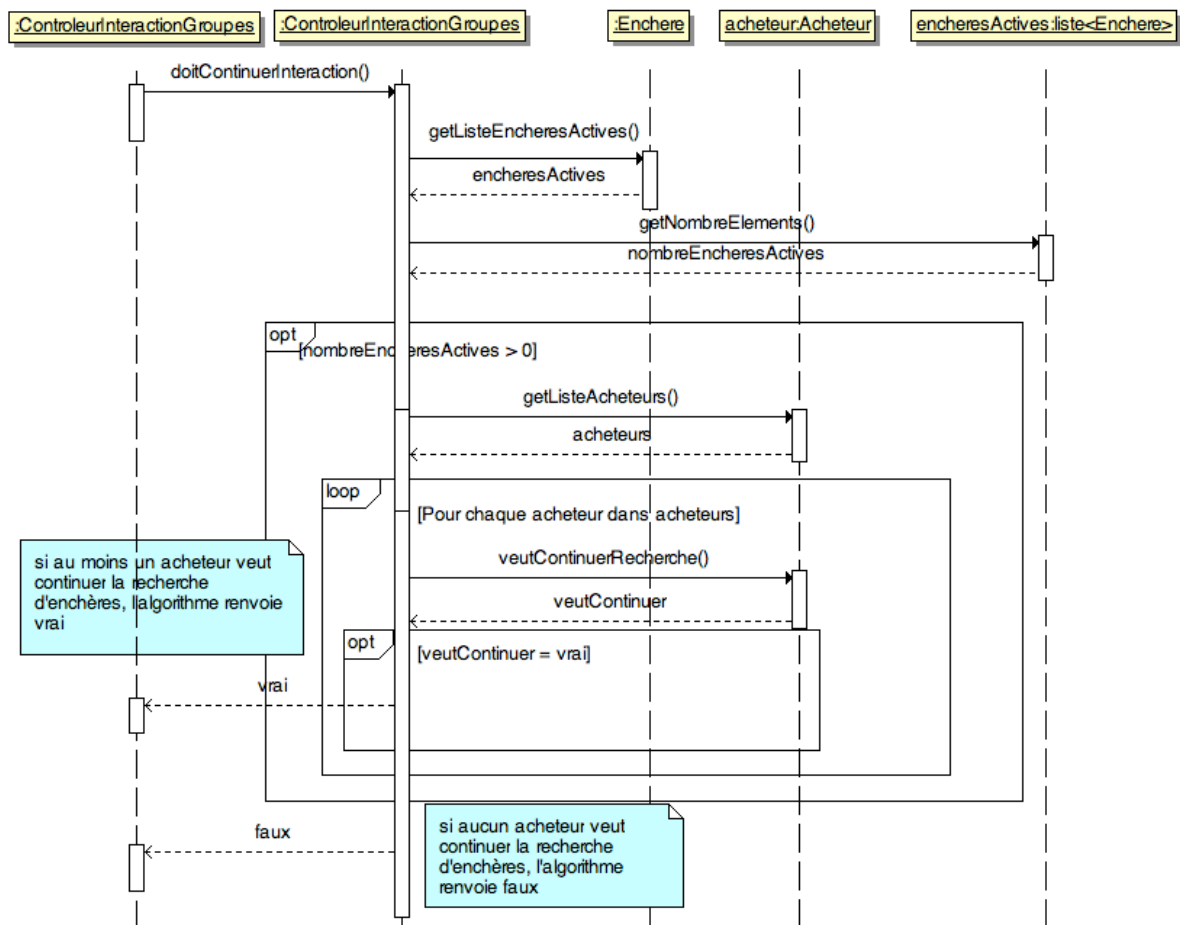


Figure 32. Diagramme de séquence pour la méthode *doitContinuerInteraction()* de la classe *ControleurInteractionGroupes*

## **Chapitre VII : Modèles de gestion de la distribution de la population**

Pour gérer la distribution de la population nous proposons deux modèles différents : un modèle dit synthétique qui considère uniquement la relation entre groupes sociaux et types d'habitat ; et un modèle dit étendu qui considère deux niveaux, le niveau mésoscopique c'est-à-dire le niveau des groupes (relation entre groupes sociaux et types d'habitat) et le niveau microscopique (relation entre ménages et logements). Le choix du modèle conditionne aussi le mécanisme d'évolution de la population et du parc de logements utilisé : dans le modèle synthétique, l'évolution agit directement au niveau des groupes ; dans le modèle étendu, l'évolution agit au niveau microscopique et ses effets sont transmis au niveau des groupes.

Dans la définition des deux mécanismes d'évolution, nous privilégions une approche globale. En effet, l'évolution de la population de ménages et du parc de logements est le produit de mécanismes complexes et difficiles à représenter par une approche individuelle (comme les règles individuelles d'évolution, typique des approches microscopiques classiques). L'évolution de la population des ménages est le résultat de processus différents, notamment le vieillissement des ménages, les changements de statut socio-économique et les mouvements migratoires. D'autre part, l'évolution du parc de logements est aussi le produit de différents processus, comme le vieillissement du bâti, les politiques d'aménagement, ainsi que des processus d'auto-construction et d'invasion des terrains. Considérer tous ces processus s'avère une tâche compliquée, très gourmande de données et de ressources informatiques. L'approche individuelle nécessiterait des simulations très complexes, difficiles de mettre en œuvre et difficiles d'interpréter.

Les deux modèles de gestion de la distribution de la population correspondent à deux implémentations différentes du contrôleur de distribution de la population (classe *ControleurDistributionPopulation* dans le package « Distribution de la Population », voir diagramme de classe ci-dessous)

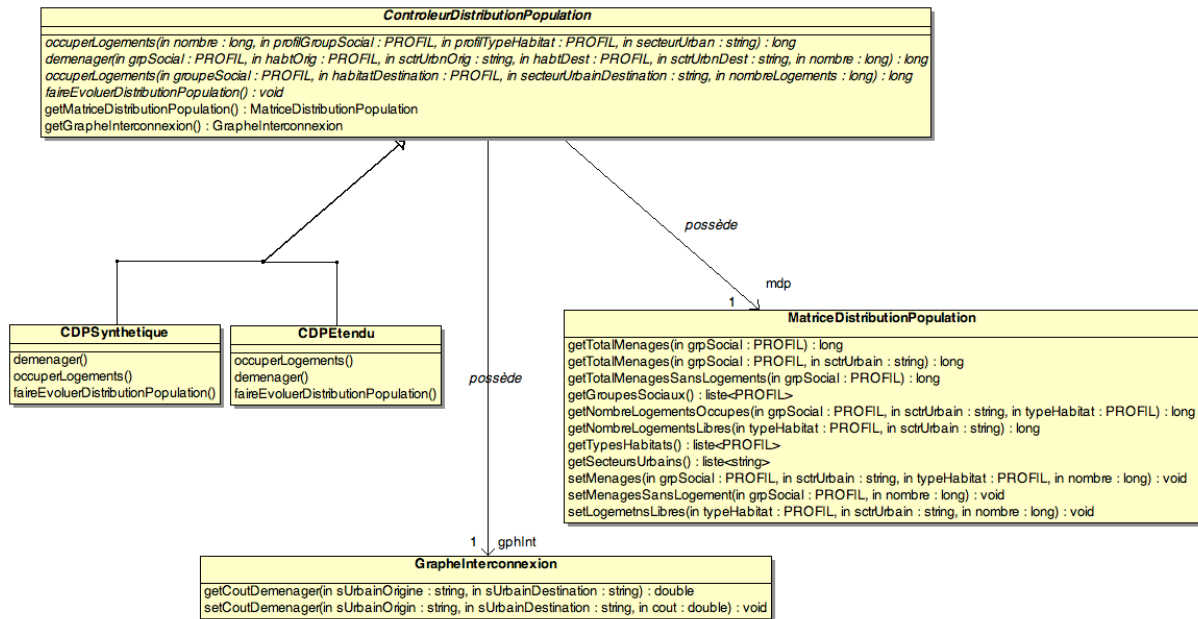


Figure 33. Diagramme de classe. Gestion de la distribution de la population

## 7.1. Modèle synthétique de gestion de la distribution de la population

### 7.1.1. Représentation de la distribution de la population

Dans le modèle synthétique de gestion de la distribution de la population, nous considérons seulement les relations entre groupes sociaux et types d'habitats. Ces relations sont représentées sous forme de matrice de distribution de la population. L'évolution de la population de ménages et du parc de logements est représentée sous forme de matrice d'évolution des groupes sociaux et de matrice d'évolution des types d'habitats.

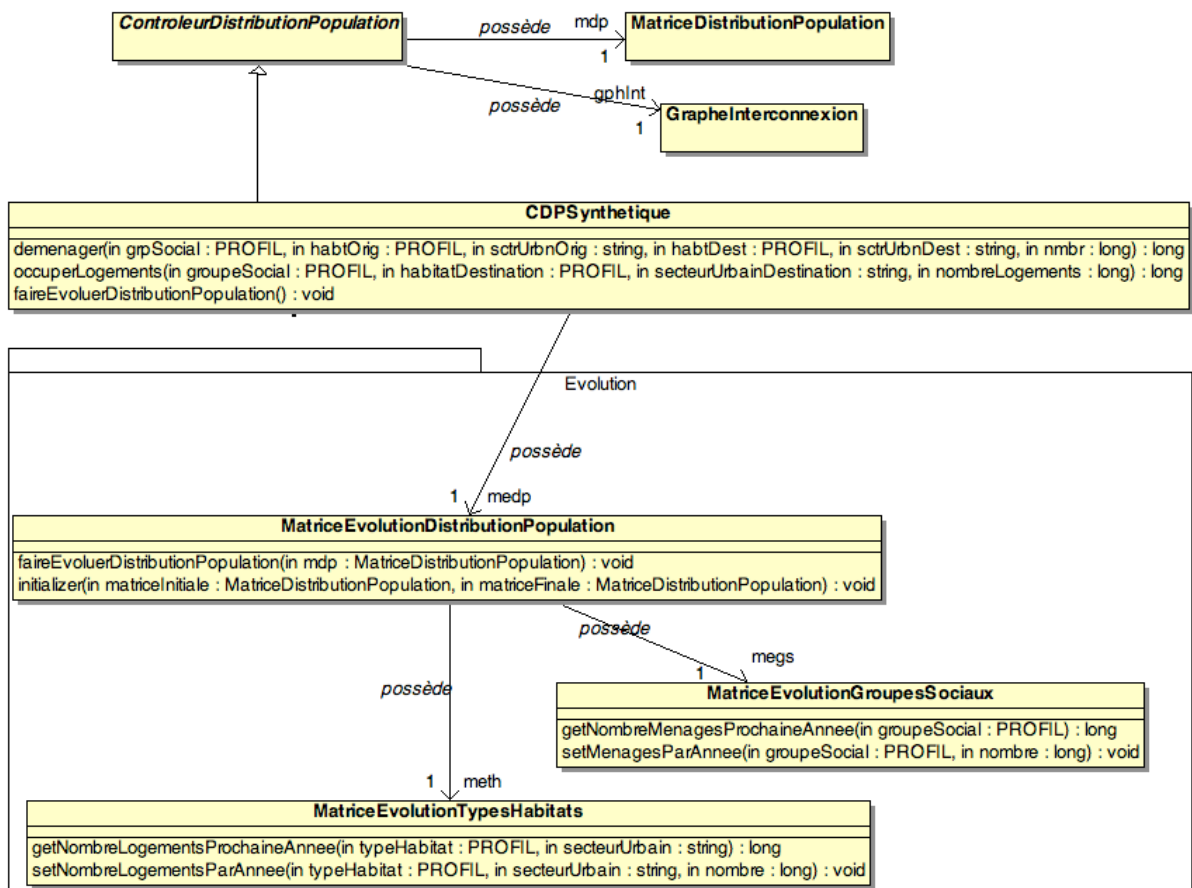


Figure 34. Diagramme de classe. Modèle synthétique de gestion de la distribution de la population

### 7.1.2. Prise en compte de la mobilité

Dans le modèle synthétique, les déménagements sont considérés comme des changements dans la relation entre groupes sociaux et types d'habitat. Deux opérations sont considérées : le déménagement, et l'occupation des logements.

Un déménagement est défini par le profil du groupe social qui déménage, l'origine (le type d'habitat d'origine et le secteur urbain d'origine), la destination (le type d'habitat de destination et le secteur urbain de destination) et le nombre  $N$  de ménages qui déménagent. Cette opération est implémentée dans la méthode *demenager* de la classe *MatriceDistributionPopulation* du package *Evolution*. L'opération de déménagement correspond à la modification des effectifs des matrices d'occupations d'origine et de destination. Dans la matrice d'origine, les logements occupés par les ménages qui déménagent sont libérés : le nombre de logements libres du type d'habitat d'origine est augmenté de  $N$ . Dans la matrice d'occupation de destination, deux actions sont aussi réalisées :

- Le nombre de logements libres du type d'habitat de destination est diminué de  $N$ .

- Le nombre de logements du type d'habitat de destination occupés par le groupe social qui déménage est augmenté de  $N$ .

L'opération d'occupation de logements (méthode *occuperLogements* de la classe *MatriceDistributionPopulation*) est une forme particulière de déménagement, dans lequel il n'y a pas d'origine, donc il n'y a pas d'actions à réaliser sur la matrice d'occupation d'origine. Cette opération est utilisée pour les déménagements des ménages qui n'ont pas de logement. Les opérations réalisées sur la matrice d'occupation de destination sont les mêmes que pour un déménagement. De plus, le nombre de ménages-sans-logement du groupe social qui déménage est diminué de  $N$ .

### 7.1.3. Evolution de la population de ménages et du parc de logements

L'hypothèse fondamentale dans l'évolution synthétique est que le nombre et le profil des groupes sociaux et des types d'habitat ne changent pas. Dans l'évolution synthétique, on ne considère que les changements dans la taille (nombre d'effectifs) des groupes sociaux et des types d'habitat. Ces changements sont exprimés sous forme de matrices d'évolution des groupes sociaux et des types d'habitat. Ces deux matrices sont construites à partir des matrices de distribution des populations initiale et finale. Cette évolution correspond à la phase d'évolution de la population simulée (1<sup>ère</sup> partie de la boucle principale) soulignée en vert dans le macro-algorithme principal de simulation ci-dessous.

Les matrices d'évolution des groupes sociaux et des types d'habitat sont construites à partir des matrices de distribution des populations réelles initiale et finale. Cette étape correspond à l'initialisation des paramètres d'évolution soulignée en jaune dans le macro-algorithme de simulation ci-dessous. La matrice de distribution de la population simulée est au départ une copie de la matrice de distribution réelle pour l'année initiale (voir étape 2 de la simulation soulignée en vert dans l'algorithme ci-dessous). L'algorithme de classification pour la formation des groupes passés en paramètre (soulignée en jaune) est utilisé pour la création des matrices de distribution des populations réelles initiale et finale.

Simulation (*population initiale réelle, population finale réelle, nombre d'années, algorithme de classification pour la formation de groupes*)

DEBUT

1. Initialiser les paramètres d'évolution de la population :

a. Créer les matrices de distribution des populations initiale et finale réelles à partir de la *population initiale réelle* et de la *population finale réelle*, en créant les groupes sociaux et les

- types d'habitats avec l'*algorithme de classification pour la formation de groupes*
- b. A partir des effectifs de chaque groupe social dans les matrices de distributions initiales réelle et finale réelle construire la matrice d'évolution de groupes sociaux
  - c. A partir des effectifs de chaque type d'habitats dans les distributions initiales réelle et finale réelle construire la matrice d'évolution de types d'habitats
2. Initialiser la *population simulée* à partir de la *population initiale réelle* :
- a. Créer la *matrice de distribution simulée* à partir de la matrice de distribution de la population initiale réelle : copier la matrice de distribution initiale réelle dans la *matrice de distribution simulée*.
3. POUR *année=1* jusqu'au *nombre d'années*
- FAIRE
- a. Evoluer la *population simulée* :
    - a. Actualiser le nombre d'effectifs de chaque groupe social dans la *matrice de distribution simulée* à partir des informations dans la matrice d'évolution de groupes sociaux
    - b. Actualiser le nombre d'effectifs de chaque type d'habitat dans la *matrice de distribution simulée* à partir des informations dans la matrice d'évolution de types d'habitats
  - b. Actualiser la répartition de la *population simulée*.
  - c. *année \_ année + 1*
- FIN POUR
4. Valider la *population simulée* contre la *population finale réelle*
- FIN

Encadré 5. Initialisation et évolution de la population simulée pour le modèle synthétique dans le macro-algorithme principal de simulation.

### 7.1.3.1. Evolution des groupes sociaux

Pour représenter l'évolution des groupes sociaux, on tient compte du changement du nombre d'effectifs entre les matrices de distribution réelle des populations initiale et finale. Ainsi, la matrice d'évolution des groupes sociaux contient le changement annuel en nombre d'effectifs nécessaires pour passer des effectifs dans la distribution initiale aux effectifs dans la distribution finale. Chaque élément dans la matrice d'évolution des groupes sociaux contient la différence (reportée sur le nombre d'années qui séparent les deux distributions) entre les effectifs des deux distributions. Par exemple, si l'on considère les matrices de distribution de la population à l'année 0 et à l'année 19 (voir encadrés 4 et 5 ci-après), pour le groupe social GS2, on obtient un changement annuel de 37 effectifs :

$$((3446 \text{ effectifs dans l'année } 19) - (2714 \text{ effectifs dans l'année } 0)) / 20 = 37 \text{ effectifs.}$$

Ce qui veut dire simplement que pour passer des 2714 effectifs dans l'année 0 aux 3446 effectifs dans l'année 19 pour le groupe GS2 on doit ajouter 37 effectifs par an. Il faut noter que le résultat de la division a été arrondi à l'entier le plus proche, ce qui induit une erreur

relativement petite par rapport à la taille de la population. Dans l'encadré 6, on présente la matrice d'évolution des groupes sociaux. Certaines valeurs dans cette matrice sont négatives : elles représentent des effectifs à supprimer annuellement.

Pour simuler le changement des effectifs de chaque groupe social d'une année à l'autre, il faut considérer deux actions différentes :

- Ajout d'effectifs : Quand le changement annuel est positif, les effectifs sont ajoutés à la liste de ménages sans logement de la matrice de distribution de la population.
- Suppression d'effectifs : Quand le changement annuel est négatif, les effectifs sont supprimés de manière aléatoire sur l'ensemble des effectifs du groupe social. Ils peuvent alors être supprimés de la liste de ménages sans logement et/ou de une ou plusieurs des matrices d'occupation. Si l'on supprime des effectifs d'une matrice d'occupation, le nombre de logements occupés par ces effectifs est ajouté aux logements libres de la matrice.

#### 7.1.3.2. Evolution des types d'habitat

Pour représenter les changements des effectifs des types d'habitats entre deux distributions de population, on tient compte des changements annuels *localisés*. Le mécanisme d'évolution des types d'habitat doit tenir compte de la localisation, parce que les types d'habitat représentent des entités spatiales. Ainsi, on détermine le changement annuel par type d'habitat et par secteur urbain. Le changement annuel en nombre d'effectifs est stocké dans la matrice d'évolution des types d'habitat. Chaque case de la matrice contient le changement annuel par type d'habitat et par secteur urbain. Par exemple, le changement annuel nécessaire pour passer des 1102 effectifs du type d'habitat TH1 dans le secteur urbain SU1 pour l'année 0 aux 876 effectifs pour l'année 19 (voir encadrés ci-dessous) est de -11 effectifs par année, c'est-à-dire qu'il faut supprimer 11 effectifs chaque année sur les 20 ans de simulation.

Dans l'évolution des effectifs des types d'habitat, il faut considérer deux opérations différentes :

- Ajout d'effectifs : Quand le changement annuel est positif, les effectifs sont ajoutés aux logements libres du type d'habitat donné dans le secteur urbain donné.
- Suppression d'effectifs : Quand le changement annuel est négatif, les effectifs sont supprimés de manière aléatoire de la matrice d'occupation du secteur urbain donné. Ils peuvent être supprimés des logements libres et/ou des logements occupés du type d'habitat à supprimer. Si l'on supprime des logements occupés, le nombre de ménages

qui occupaient ces logements est ajouté aux listes de ménages sans logement des groupes sociaux correspondants.

<b>Liste de type d'habitats :</b>					
Type d'habitat	TH1	TH2	TH3	TH4	TH5
Effectifs	6317	4196	5146	6151	3798

<b>Liste de groupes sociaux :</b>						
Groupe social	GS1	GS2	GS3	GS4	GS5	GS6
Effectifs	2210	2714	5157	3620	5569	4974

<b>Liste de ménages sans logements :</b>						
Groupe social	GS1	GS2	GS3	GS4	GS5	GS6
Effectifs	50	35	10	25	32	19

**Matrices d'occupation :**

**1. Secteur urbain SU1**

	GS1	GS2	GS3	GS4	GS5	GS6	Log. Libres	Totaux
TH1	86	122	186	199	296	145	68	1102
TH2	205	153	306	271	440	70	86	1531
TH3	229	443	655	431	852	414	69	3093
TH4	257	303	505	453	631	498	160	2807
TH5	155	201	275	192	259	415	896	2393
<i>Totaux</i>	<i>932</i>	<i>1222</i>	<i>1927</i>	<i>1546</i>	<i>2478</i>	<i>1542</i>	<i>1279</i>	<i>10926</i>

**2. Secteur urbain SU2**

	GS1	GS2	GS3	GS4	GS5	GS6	Log. Libres	Totaux
TH1	537	52	1461	21	2565	367	212	5215
TH2	204	51	1225	674	469	9	33	2665
TH3	328	155	438	1083	15	34	0	2053
TH4	159	137	12	46	0	2979	11	3344
TH5	0	1062	84	225	10	24	0	1405
<i>Totaux</i>	<i>1228</i>	<i>1457</i>	<i>3220</i>	<i>2049</i>	<i>3059</i>	<i>3413</i>	<i>256</i>	<i>14682</i>

Encadré 6. Exemple de matrice de distribution de la population pour l'année initiale (année 0) de simulation

<b>Liste de type d'habitats :</b>					
Type d'habitat	TH1	TH2	TH3	TH4	TH5
Effectifs	9965	3715	5142	5101	4344

<b>Liste de groupes sociaux :</b>						
Groupe social	GS1	GS2	GS3	GS4	GS5	GS6
Effectifs	2194	3446	5849	3422	7981	3680

<b>Liste de ménages sans logements :</b>						
Groupe social	GS1	GS2	GS3	GS4	GS5	GS6
Effectifs						



Effectifs	19	31	10	12	50	4		
-----------	----	----	----	----	----	---	--	--

**Matrices d'occupation :**

**1. Secteur urbain SU1**

	GS1	GS2	GS3	GS4	GS5	GS6	Log. Libres	Totaux
TH1	51	203	205	138	183	5	91	876
TH2	42	266	575	366	455	96	101	1901
TH3	290	446	808	223	617	465	99	2948
TH4	352	294	483	566	571	831	241	3338
TH5	114	41	408	276	372	240	874	2325
Totaux	849	1250	2479	1569	2198	1637	1406	11388

**2. Secteur urbain SU2**

	GS1	GS2	GS3	GS4	GS5	GS6	Log. Libres	Totaux
TH1	625	37	2549	14	4989	516	359	9089
TH2	220	30	551	237	730	3	43	1814
TH3	399	261	127	1360	5	42	0	2194
TH4	82	185	3	23	0	1457	13	1763
TH5	0	1652	130	207	9	21	0	2019
Totaux	1326	2165	3360	1841	5733	2039	415	16879

Encadré 7. Exemple de matrice de distribution de la population pour l'année finale (année 19) de simulation

**Matrice d'évolution annuelle des effectifs par type d'habitats et par secteur urbain**

	TH1	TH2	TH3	TH4	TH5
SU1	-11	19	-7	27	-3
SU2	194	-43	7	-79	31

**Matrice d'évolution annuelle des effectifs pour chaque groupe social**

Groupe social	GS1	GS2	GS3	GS4	GS5	GS6
Effectifs	-1	37	35	-10	121	-65

Encadré 8. Exemple de matrices synthétiques d'évolution de la distribution de population

## 7.2. Modèle étendu de gestion de la distribution de la population

Dans le modèle synthétique, les groupes sociaux et les types d'habitat sont considérés d'une manière relativement statique : il n'y a que le nombre d'effectifs qui évolue, le nombre et le profil des groupes sociaux et des types d'habitat ne changent pas. Cette première approche suppose une stabilité dans le temps des groupes sociaux et des types d'habitat. Dans un deuxième temps, nous voulons considérer l'impact de l'évolution au niveau microscopique en ce qui concerne la structure des groupes sociaux et des types d'habitat. Il est donc nécessaire

d'évacuer l'hypothèse de stabilité des groupes sociaux et des types d'habitats. De cette manière, nous pouvons mesurer l'impact des évolutions microscopiques dans la structure des groupes sociaux et des types d'habitats, c'est-à-dire, les changements dans le nombre et les profils des groupes sociaux et des types d'habitat.

Nous proposons deux systèmes complémentaires pour représenter la relation entre l'évolution au niveau microscopique et le niveau des groupes. Le premier mécanisme modélise l'évolution de la population des ménages et du parc de logements par des règles générales d'évolution. Le deuxième mécanisme permet la reconstitution des groupes sociaux et des types d'habitat par un processus de classification automatique effectué sur les ménages et les logements. Dans l'algorithme ci-dessous, nous avons souligné ces étapes et les paramètres nécessaires pour les accomplir. En jaune, nous avons souligné le paramètre et l'étape de construction des règles d'évolution. En vert, nous avons souligné l'initialisation de la population simulée et son évolution, ainsi que l'algorithme de classification utilisé dans ces deux étapes pour la formation de groupes.

Simulation (*population initiale réelle, population finale réelle, nombre d'années, algorithme de classification pour la formation de groupes, algorithme de classification pour la construction des règles d'évolution*)

DEBUT

1. Initialiser les paramètres d'évolution de la population :

- a. Créer les matrices de distribution des populations initiale et finale réelles à partir de la *population initiale réelle* et de la *population finale réelle*, en créant les groupes sociaux et les types d'habitats avec l'*algorithme de classification pour la construction des règles d'évolution*
- b. A partir des effectifs de chaque groupe social dans les matrices de distributions initiales réelle et finale réelle construire les règles d'évolution pour les groupes sociaux
- c. A partir des effectifs de chaque type d'habitats dans les distributions initiales réelle et finale réelle construire les règles d'évolution pour les types d'habitats

2. Initialiser la *population simulée* à partir de la *population initiale réelle* :

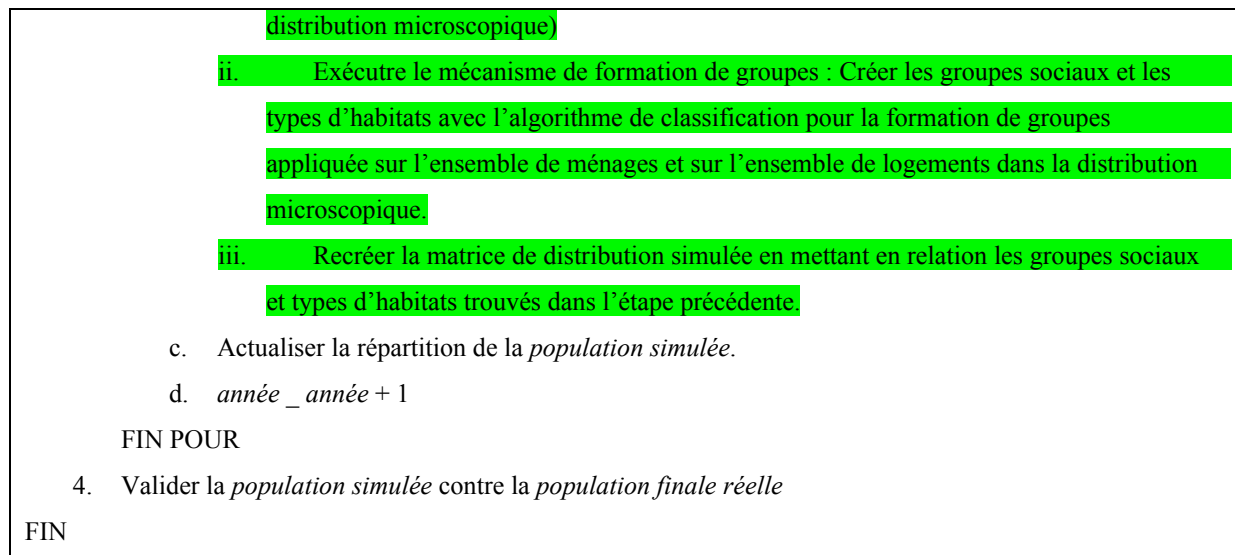
- a. Recréer les matrices de distribution des population initial et finale réelles à partir de la *population initiale réelle* et de la *population finale réelle*. MAIS CETTE FOIS-CI, en créant les groupes sociaux et les types d'habitats avec l' *algorithme de classification pour la formation de groupes*
- b. Créer la *matrice de distribution simulée* à partir de la matrice de distribution de la population initiale réelle : copier la matrice de distribution initiale réelle dans la *matrice de distribution simulée*.

3. POUR *année=1* jusqu'au *nombre d'années*

FAIRE

a. Evoluer la *population simulée* :

- i. Exécuter les règles d'évolution au niveau microscopique (Changements dans la



Encadré 9. Initialisation et évolution de la population simulée pour le modèle étendu dans le macro-algorithme principal de simulation.

### 7.2.1. Représentation de la distribution de la population

Dans le modèle étendu, nous considérons une double représentation de la distribution de la population. D'une part, au niveau mésoscopique, nous considérons une matrice de distribution de la population qui exprime la relation entre groupes sociaux et types d'habitat. Et d'autre part, nous considérons la distribution microscopique de la population (liste de triplets ménage-logement-secteur urbain, liste de ménages-sans-logement et liste de logements-libres), qui représente la relation entre les ménages et les logements. Cette représentation est implémentée par les classes présentées dans le diagramme ci-dessous.

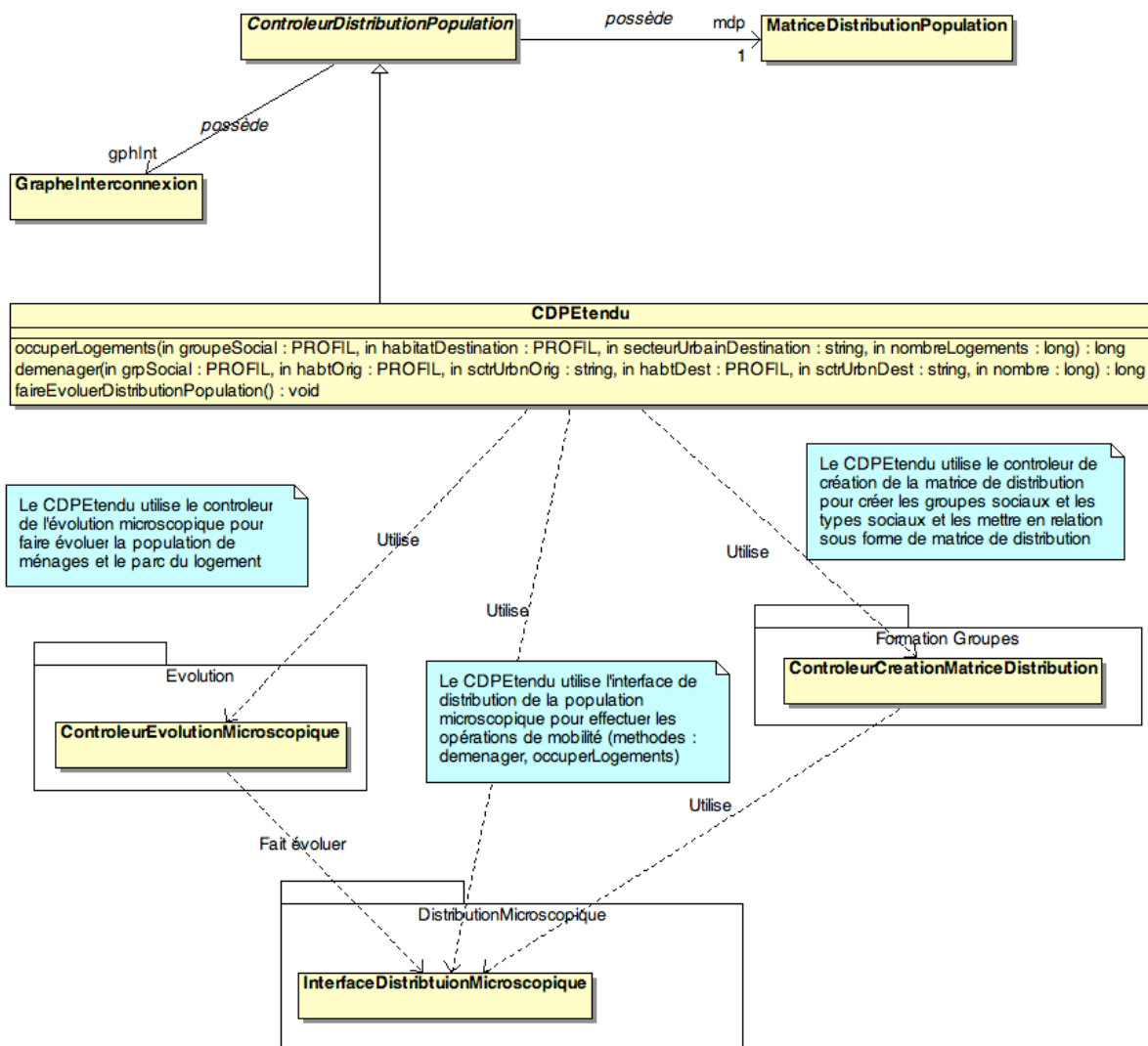


Figure 35. Diagramme de classe. Modèle étendu de gestion de la distribution de la population

L'évolution de la distribution de la population est effectuée par l'algorithme présenté ci-dessous qui combine le mécanisme d'évolution au niveau microscopique avec le mécanisme d'actualisation de la matrice de distribution de population (reconstruction des groupes sociaux et types d'habitat et leur mise en relation). Dans la section suivante, nous présentons le mécanisme d'évolution au niveau microscopique et ensuite nous présentons le mécanisme d'actualisation de la matrice de distribution de la population.

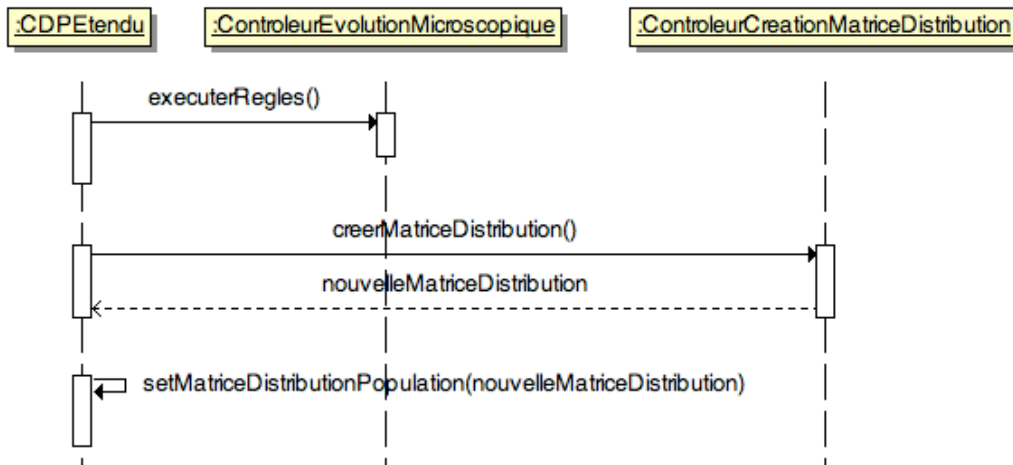


Figure 36. Diagramme de séquence. Actualisation de la matrice de distribution de la population dans le modèle étendu.

### 7.2.2. Prise en compte de la mobilité

Dans le système étendu, les opérations de déménagement et d'occupation des logements sont prises en compte à la fois aux niveaux mésoscopique et au niveau microscopique. Le mécanisme de mobilité au niveau mésoscopique est le même que dans le système synthétique. Le mécanisme de mobilité au niveau microscopique est implémenté par les méthodes *demenager* et *occuperLogements* de la classe *ControleurDistributionMicroscopique*, qui sont accessibles par l'intermédiaire de l'interface à la distribution microscopique. Il existe deux implémentations différentes du mécanisme de mobilité au niveau microscopique qui diffèrent quant à la localisation du processus de décision de déménager : la première implémentation ne considère pas le ménage comme participant au processus de déménager ; la deuxième implémentation considère une interaction « faible » du ménage dans le processus de décision. Dans le premier cas, les ménages suivent la demande des groupes sociaux de déménager. Dans le deuxième cas, le ménage a une possibilité de refuser de déménager en fonction du calcul d'une probabilité de refus.

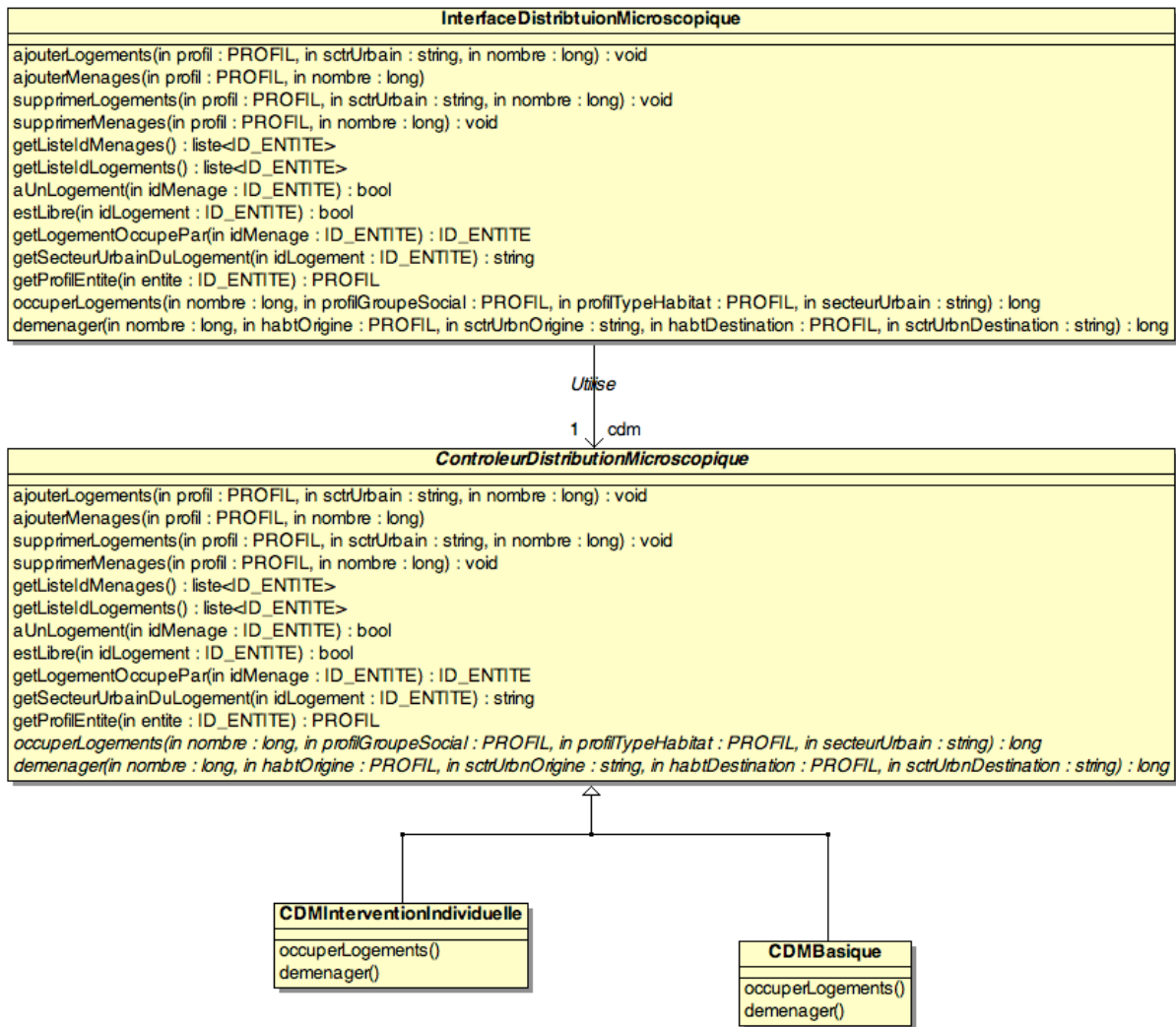


Figure 37. Diagramme de classe. Package distribution microscopique de la population.

### 7.2.2.1. Décision de déménager au niveau mésoscopique

Dans ce cas, la décision de déménager est prise au niveau mésoscopique selon les mécanismes présentés dans le chapitre VI. Le déménagement est demandé par les groupes sociaux. Le message de cette demande atteint le niveau microscopique par l'interface de distribution microscopique qui fait déménager le nombre de ménages voulus. Le nombre de ménages qui déménagent au niveau microscopique est exactement le nombre demandé par les groupes sociaux. Ce mécanisme est implémenté dans la classe *CDMBasique* qui est une spécialisation de la classe abstraite *ControleurDistributionMicroscopique*. Les deux opérations « déménager » et « occuper logements » sont implémentés dans cette classe.

Dans le cas de l'opération de déménagement, on choisit les  $N$  ( $N$  étant le nombre de ménages qui déménagent) couples de ménage-logement qui sont dans le secteur urbain d'origine et dont le profil du ménage est le plus proche possible du profil du groupe social qui déménage,

et le profil du logement qui est le plus proche possible du profil du type d'habitat d'origine. Ensuite, on sélectionne les  $N$  logements dans le secteur urbain de destination dont le profil est le plus proche possible du profil du type d'habitat de destination. Finalement, on libère les  $N$  logements dans le secteur urbain d'origine et on met en relation les  $N$  ménages avec les  $N$  logements dans le secteur de destination. La proximité entre deux profils est calculée à partir de la distance de Hamming (cf. Annexe III, Calculs de distances entre profils).

Dans le cas de l'opération d'occupation de logements, on choisit d'abord les  $N$  ménages-sans-logement qui sont les plus proches du profil du groupe social qui déménage. Ensuite, on choisit les  $N$  logements-libres dans le secteur urbain de destination qui sont les plus proches du profil du type d'habitat de destination. Finalement, on crée un lien entre les ménages et les logements choisis.

#### 7.2.2.2. Décision de déménager au niveau mésoscopique avec une participation « faible » du niveau microscopique

La deuxième implémentation du mécanisme de mobilité au niveau microscopique est réalisée dans la classe *CDMInterventionIndividuelle*. Dans cette classe, les ménages et les logements qui interviennent dans les opérations de déménagement ou d'occupation de logements sont choisis avec le même mécanisme implémenté dans le *CDMBasique*. La seule différence est que dans l'implémentation du *CDMInterventionIndividuelle*, chaque ménage détient la possibilité de refuser de déménager. En conséquence, le nombre de ménages qui déménagent réellement est inférieur ou égal au nombre demandé par le groupe social. Un ménage accepte ou refuse de déménager selon une probabilité de refus. Cette probabilité est fonction de la proximité entre le profil du ménage et le profil du groupe social : plus le profil du ménage est différent du profil du groupe social, plus la probabilité de refuser de déménager est grande.

### **7.2.3. Evolution au niveau microscopique**

Pour représenter l'évolution de la population des ménages et du parc de logements, nous proposons un système basé sur la définition de règles générales d'évolution qui agissent au niveau microscopique. Ces règles d'évolution sont construites à partir de deux matrices de distribution de la population réelle qui représentent la distribution de la population de la ville pour deux années différentes.

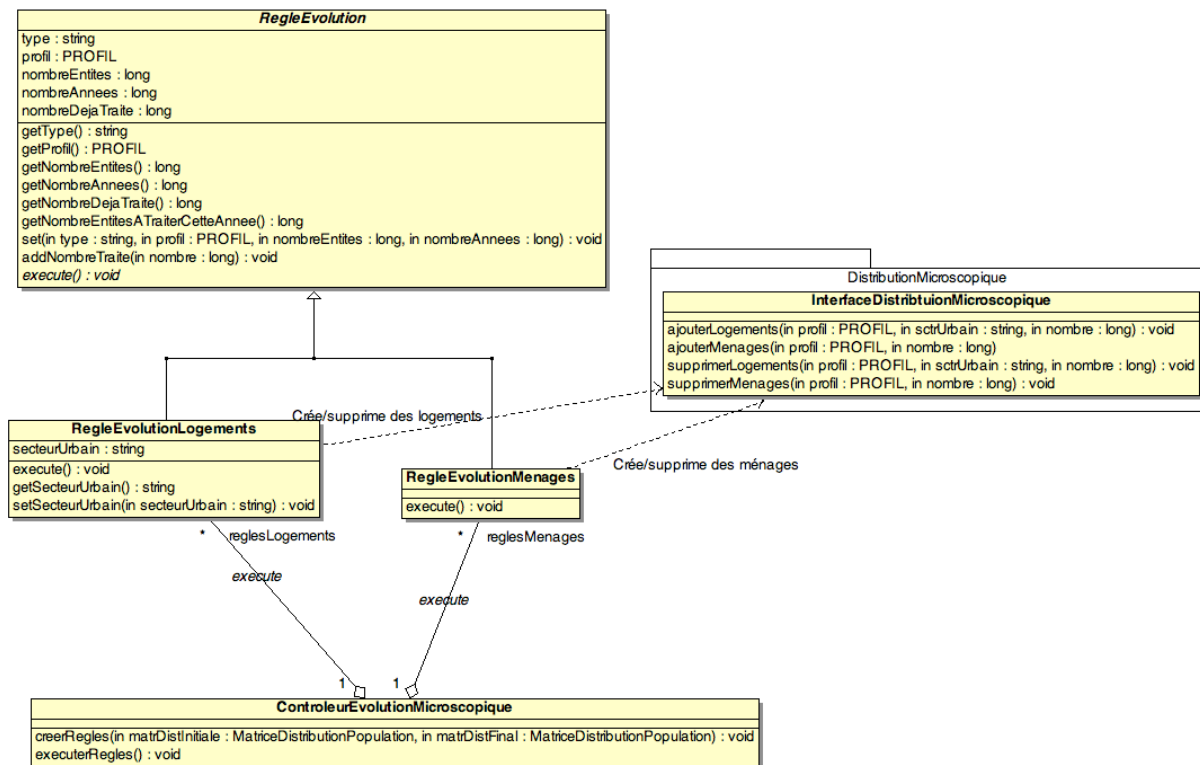


Figure 38. Diagramme de classe. Évolution au niveau microscopique

### 7.2.3.1. Règles d'évolution pour les ménages

Les règles d'évolution permettent la création ou la destruction d'entités avec un profil donné. Dans le cas général (cas de règles de ménages) une règle d'évolution est définie par un profil, un type, un nombre  $N$  d'entités à traiter, un nombre  $A$  d'années d'exécution, le nombre  $T$  d'entités déjà traitées par la règle et le nombre d'exécutions  $E$  déjà effectuées. Le type est soit suppression soit création.

Exemple de règle d'évolution des ménages : Si l'on considère la liste de groupes sociaux présentée dans l'encadré 6 et que l'on veut faire disparaître le groupe social GS3 sur une période de 20 ans, la règle d'évolution correspondante est :

Règle\_GS3 : {type=suppression ; profil= <profil(GS3)> ; N = 5157; A=20 ; T=0 ; E=0}

Le nombre  $N$  est égal aux effectifs du groupe social GS3.  $T$  et  $E$  sont initialisés à 0, mais changent à chaque exécution de la règle.

### 7.2.3.2. Règle d'évolution pour les logements

En plus des définitions de base, elle contient aussi le secteur urbain où la règle doit s'appliquer.



Exemple de règle d'évolution d'habitat : Si l'on considère la matrice d'occupation l'encadré 6 et que l'on veut faire disparaître le type d'habitat TH1 sur une période de 20 ans, la règle d'évolution correspondante est :

Règle\_TH1\_SU1 : {type=suppression ; profil= <profil(TH1)> ; N = 1102 ; A=20 ; Secteur\_Urbain = SU1 ; T=0 ; E=0}

Le nombre  $N$  est égal à la somme des effectifs du type d'habitat TH1 dans le secteur urbain SU1 (somme des valeurs dans la première ligne de la matrice d'occupation de SU1 dans l'encadré 6).  $T$  et  $E$  sont initialisés à 0, mais changent à chaque exécution de la règle.

### 7.2.3.3. Exécutions d'une règle d'évolution

Si le type de la règle est création, au bout de  $A$  exécutions,  $N$  entités du profil donné sont ajoutées à la distribution réelle de la population. Si les entités à créer sont des ménages, elles sont ajoutées à la liste des ménages-sans-logement de la distribution microscopique de la population. Si les entités à créer sont des logements, elles sont ajoutées à la liste des logements libres *dans le secteur urbain* défini par la règle.

Si le type de la règle est suppression, au bout de  $A$  exécutions, les  $N$  entités les plus proches du profil donné<sup>10</sup> sont supprimées de la distribution réelle de population.

- a. Si les entités à supprimer sont des ménages, elles sont choisies parmi les listes des ménages-avec-logement et des ménages-sans-logement. Dans le cas où l'on supprime des ménages de la liste des ménages-avec-logement, les logements qu'ils occupaient sont déplacés vers la liste des logement-libres de la distribution microscopique de la population.
- b. Si les entités à supprimer sont des logements, elles sont choisies parmi les logements occupés et les logements libres *dans le secteur urbain* défini par la règle. Dans le cas où l'on supprime des logements de la liste des logements occupés, les ménages qui les occupaient sont déplacés vers la liste des ménages-sans-logement de la distribution microscopique de la population.

### 7.2.3.4. Nombre d'entités traitées lors d'UNE exécution d'une règle d'évolution

Ce nombre est égal à l'entier le plus proche du quotient  $N/A$ . A chaque exécution de la règle, le nombre  $T$  est augmenté du nombre d'entités exécutées. Il existe trois cas de figure :

---

<sup>10</sup>La distance entre profils utilisée par défaut est la distance de Hamming (cf. Annexe II, Calculs des distances entre profils).

- a.  $N$  est multiple de  $A$  : Dans ce cas, le nombre d'entités traitées chaque année est exactement égal à  $N/A$ .
- b.  $N$  n'est pas multiple de  $A$  et  $N$  est supérieur à  $A/2$  : A cause de l'arrondi, le produit  $A*N/A$  est soit inférieur soit supérieur à  $N$ . Dans le premier cas, le nombre d'entités non traitées est reporté à la dernière exécution. Dans le deuxième cas, des entités sont traitées jusqu'à ce que le nombre  $T$  atteigne  $N$ . Des exécutions ultérieures ne traitent plus aucune entité.
- c.  $N$  est inférieur à  $A/2$  : Cas dans lequel  $N/A$  est égal à zéro. Au lieu de reporter le traitement de toutes les entités à la dernière exécution, on fait un traitement « probabiliste ». Dans ce cas, UNE entité est traitée à chaque exécution de la règle selon une probabilité d'exécution. Cette probabilité dépend du quotient entre le nombre d'exécutions déjà effectuées et le nombre  $A$ . La probabilité de traiter une entité croît lorsque le nombre d'exécutions déjà effectuées s'approche du nombre  $A$  d'exécutions. De la même manière que dans le cas **b**, de nouvelles entités sont traitées tant que  $T$  est inférieur à  $N$ . Dans la dernière exécution, un nombre égal à «  $N - T$  » d'entités est traité.

#### 7.2.3.5. Construction et exécution des règles d'évolution

Nous adoptons une approche globale pour représenter l'évolution de la population et du parc de logement. Cette approche est contraire à l'approche individuelle classique, utilisée notamment en microsimulation (Orcutt *et al.*, 1976 ; Clarke & Wilson, 1986 ; Caldwell & Keister, 1996 ; Shaw, 2000 ; Holm & Sanders, 2007 ; Antcliff, 1993). Nous ne considérons pas d'événements individuels tels que naissance, mariage ou mort. Au contraire, nous considérons des tendances globales qui décrivent l'évolution de la population et du parc de logement de façon synthétique.

La construction des règles d'évolution se fait à partir du nombre d'années de simulation  $A$  ainsi que de deux matrices de distribution des populations réelles correspondant respectivement à l'année initiale et à l'année finale de la période de simulation. L'objectif est de faire graduellement disparaître la distribution initiale et apparaître la distribution finale. Ainsi, on crée des règles de suppression pour faire disparaître la distribution initiale et des règles de création pour faire apparaître la distribution finale.

##### 7.2.3.5.1. Règles de suppression (disparition de la distribution initiale)

- a. Pour traiter la disparition des groupes sociaux de la distribution initiale, on crée une règle de suppression de ménages par groupe social. Le profil de la règle est le

profil du groupe ; le nombre  $N$  de ménages à traiter est égal au nombre d'effectifs dans le groupe ; le nombre d'années est  $A$ .

Au total, on crée  $SI$  règles de suppression de ménages, où  $SI$  est le nombre de groupes sociaux dans la distribution initiale.

- b.** Pour traiter la disparition des types d'habitat de la distribution initiale, on crée une règle de suppression de logements par type d'habitat et par secteur urbain. Le profil de la règle est le profil du type d'habitat ; le nombre  $N$  de logements à traiter est égal au nombre d'effectifs du type d'habitat dans le secteur urbain donné ; le nombre d'années d'exécution de la règle est  $A$ .

Au total, on crée  $UxHI$  règles de suppression de logements, où  $U$  est le nombre de secteurs urbains et  $HI$  le nombre de types d'habitats dans la distribution initiale.

#### 7.2.3.5.2. Règles de création (création de la distribution finale)

- a.** Pour traiter l'apparition de groupes sociaux dans la distribution finale, on crée une règle de création de ménages par groupe social. Le profil de la règle est le profil du groupe ; le nombre  $N$  de ménages à traiter est égal au nombre d'effectifs dans le groupe ; le nombre d'années est  $A$ .

Au total, on crée  $S2$  règles de création de ménages, où  $S2$  est le nombre de groupes sociaux dans la distribution finale.

- b.** Pour traiter l'apparition des types d'habitat de la distribution finale, on crée une règle de création de logements par type d'habitat et par secteur urbain. Le profil de la règle est le profil du type d'habitat ; le nombre  $N$  de logements à traiter est égal au nombre d'effectifs du type d'habitat dans le secteur urbain donné ; le nombre d'années d'exécution de la règle est  $A$ .

Au total on crée  $UxH2$  règles de suppression de logements, où  $U$  est le nombre de secteurs urbains et  $H2$  le nombre de types d'habitat dans la distribution finale.

Au total on crée  $SI+S2$  règles d'évolution des groupes sociaux et  $Ux(HI+H2)$  règles d'évolution des types d'habitat.

Pour reproduire l'évolution de la population des ménages et du parc de logements, on commence avec la distribution microscopique de la population initiale. A chaque année de simulation, on exécute chaque règle d'évolution, ce qui modifie la distribution microscopique de la population et qui a comme effet de faire disparaître graduellement les profils de la distribution initiale et de faire apparaître graduellement les profils de la distribution finale.

## 7.2.4. Reconstitution des groupes sociaux et des types d'habitats

Au début de chaque pas de simulation, on exécute les règles d'évolution sur la population de ménages et le parc de logements. Les changements produits par ces règles au niveau microscopique doivent se transmettre au niveau mésoscopique. La transmission de ces changements est effectuée par deux processus de classification automatique sur les ménages et sur les logements qui permettent de construire de nouveaux groupes sociaux et de nouveaux types d'habitat. Ensuite, les nouveaux groupes sociaux et les nouveaux types d'habitats sont mis en relation sous forme d'une nouvelle matrice de distribution de la population. Ce processus de création de la nouvelle matrice de distribution de la population est effectué par deux méthodes complémentaires : *creerMatriceDistribution* et *actualiserMatriceDistribution* de la classe *ControlleurCreationMatriceDistribution* (voir ci-après les diagrammes de séquence de ces méthodes).

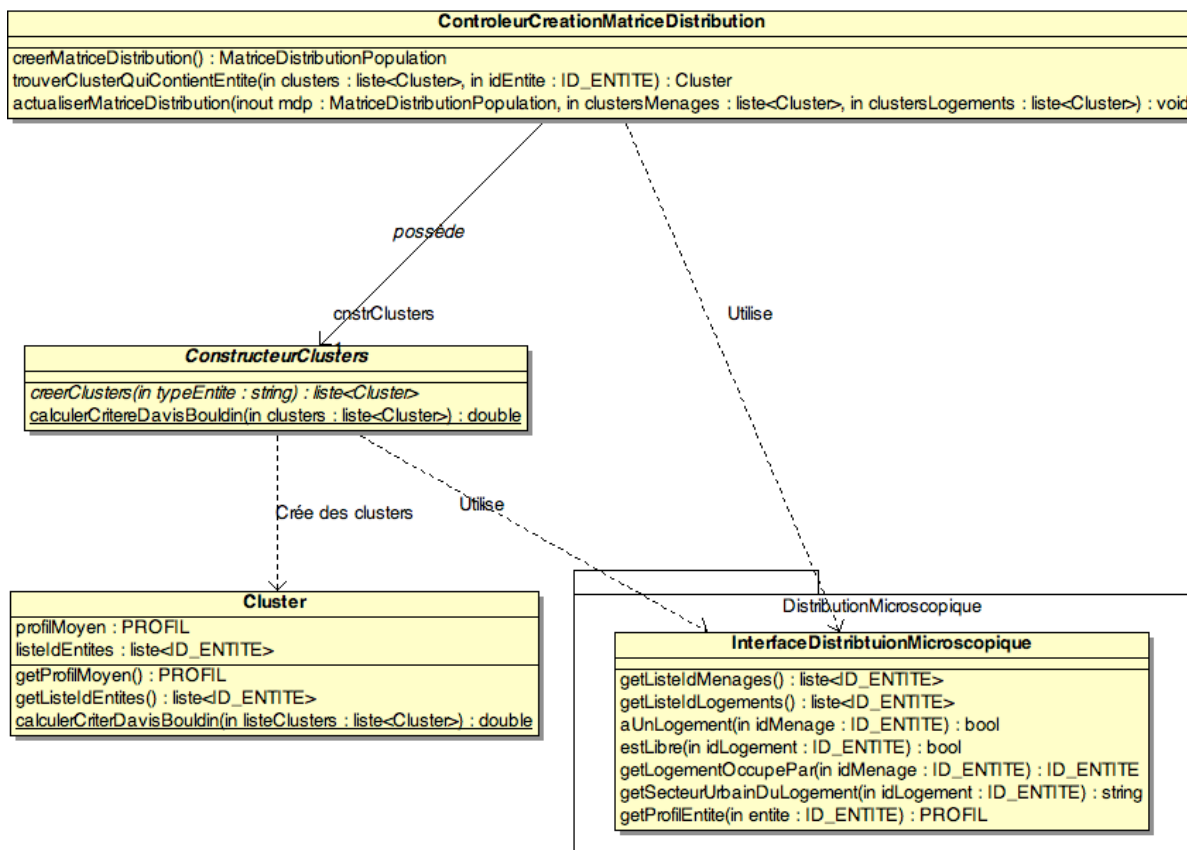


Figure 39. Diagramme de classe. Formation des groupes sociaux et des types d'habitats

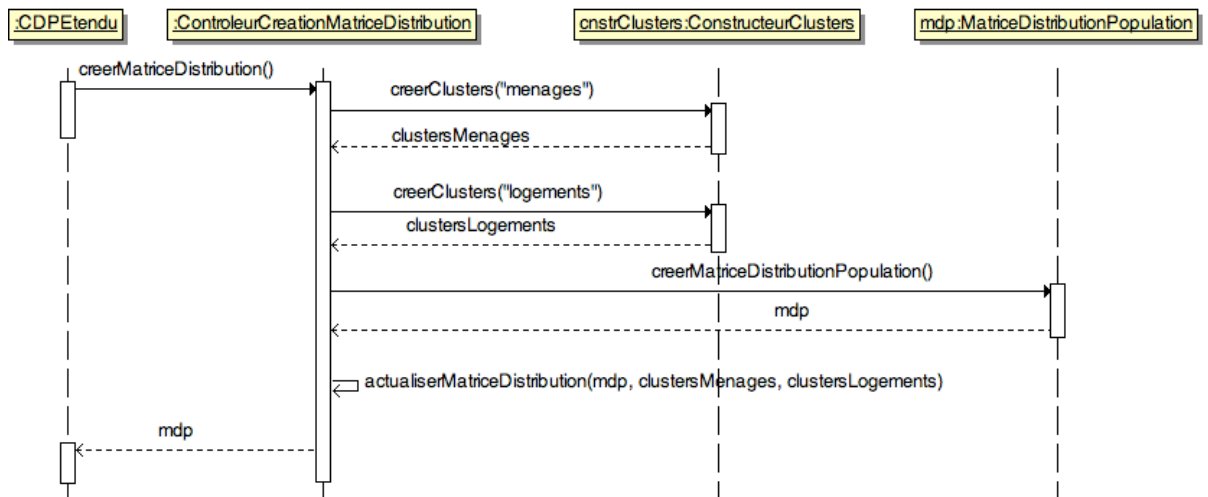


Figure 40. Diagramme de classe. Création de la matrice de distribution de la population

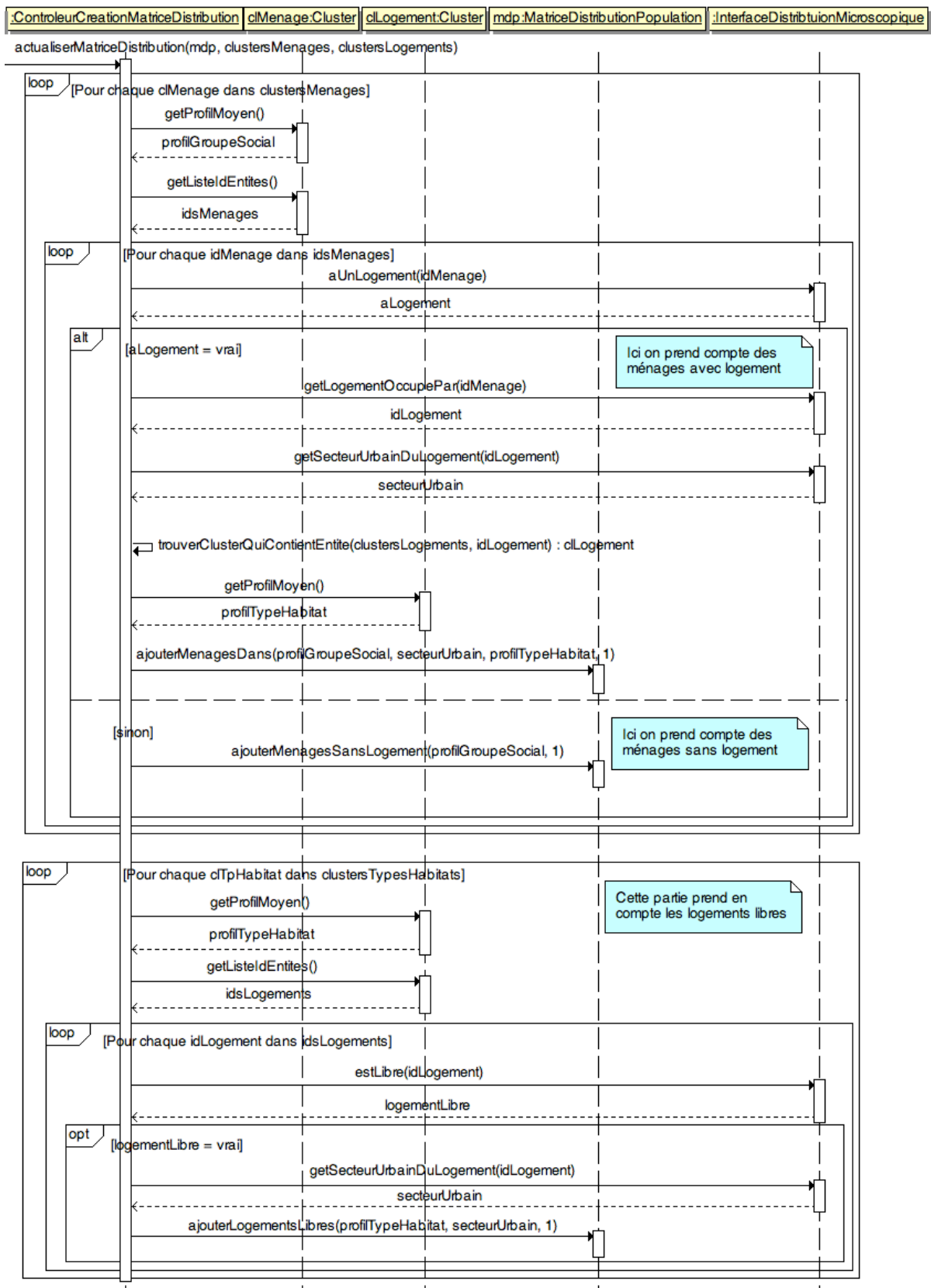


Figure 41. Diagramme de séquence. Création de la matrice de distribution de la population : méthode *actualiserMatriceDistribution* de la classe *ControleurCreationMatriceDistribution*.

### 7.2.5. Histoire des groupes sociaux et des types d'habitat

À la différence du modèle synthétique, dans le modèle étendu, le nombre et la description (profil moyen) des groupes sociaux et des types d'habitat ne sont pas fixes. Le changement dans le nombre implique évidemment l'apparition ou la disparition de certains groupes sociaux et de certains types d'habitat. Le changement dans le profil rend difficile la « traçabilité » de chaque groupe social et type d'habitat entre deux années de simulation. Il faut pouvoir mettre en relation les ensembles de groupes sociaux et de types d'habitat entre plusieurs années consécutives. Nous proposons un système basé sur la différence entre profils qui permet de trouver les correspondances et différences entre les ensembles de groupes sociaux et de types d'habitat de deux années consécutives.

Le mécanisme proposé consiste d'abord, à construire deux matrices de différences de profils entre les deux années consécutives. La première matrice contient les différences de profils entre les deux ensembles de groupes sociaux. La deuxième matrice contient les différences de profils entre les deux ensembles de types d'habitat. À partir de ces deux matrices, on détermine des équivalences d'une année à une autre. L'idée est de trouver une correspondance pour chaque groupe social et type d'habitat de la première année parmi les groupes sociaux et types d'habitat de la deuxième. Pour calculer la différence entre deux profils, nous utilisons la distance de Hamming (cf. Annexe III, Calculs de distances entre profils).

Le processus pour déterminer les correspondances entre types d'habitats et groupes sociaux est le même :

Trouver correspondances entre groupes sociaux et types d'habitats

Entrée : Matrice de différences entre profils (de groupes sociaux ou de types d'habitats, selon le cas)

Sorties : Liste de couples, liste de profils sans correspondance pour la première année, liste de profils sans correspondance pour la deuxième année.

DEBUT

8. Choisir le couple de profils dont la différence est minimale.
9. Si aucun des profils n'est dans la liste de couples, ajouter le couple choisi à la liste des couples.
10. marquer le couple choisi comme déjà analysé.
11. S'il reste des couples sans marquer revenir à 1.
12. Pour chaque profil de la première année
  - a. Si le profil n'est pas dans la liste des couples, alors ajouter le profil à la liste des profils sans correspondance de la première année
13. Pour chaque profil de la deuxième année

- a. Si le profil n'est pas dans la liste de couples alors ajouter le profil à la liste de profils sans correspondance de la deuxième année

FIN

Encadré 10. Algorithme de mise en relation des groupes sociaux et des types d'habitats de deux années consécutives.

Les groupes sociaux ou types d'habitat (selon le cas) dont les profils sont dans la liste de profils sans correspondance de la première année sont considérés comme disparus dans la nouvelle année. Les groupes sociaux ou types d'habitat (selon le cas) dont les profils sont dans la liste de profils sans correspondance de la deuxième année sont considérés comme apparus dans la nouvelle année.

À partir du traçage des groupes nous construisons à la fin de la simulation des diagrammes « historiques » des groupes sociaux et des types d'habitat. Pour chaque année de simulation, nous construisons des cartes d'occupation des groupes sociaux et des cartes de distribution des types d'habitats par année de simulation (Voir exemples ci-dessous).

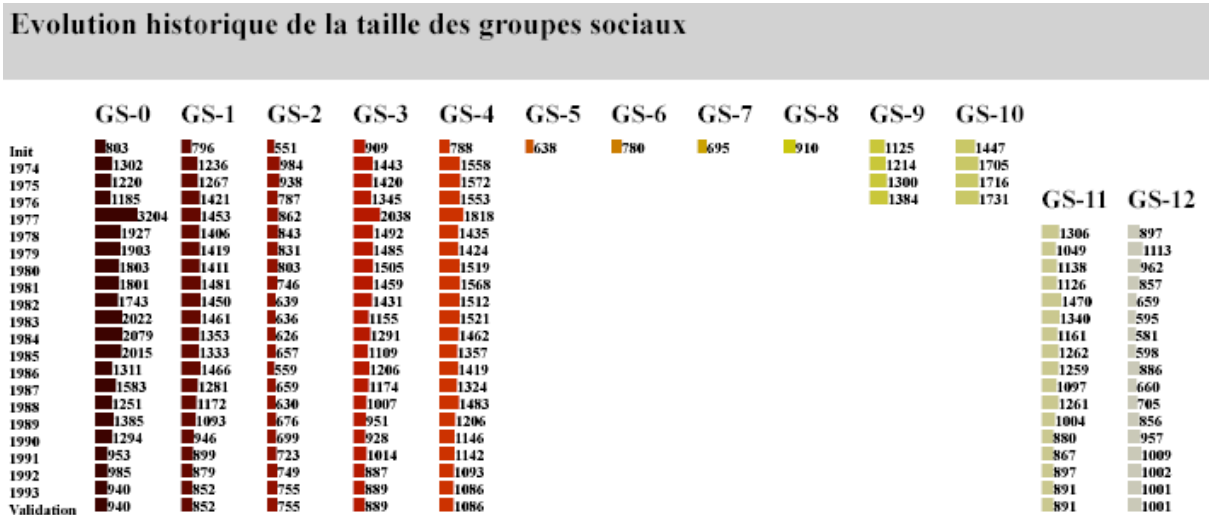


Figure 42. Exemple de diagramme historique pour les groupes sociaux. Chaque colonne de barres représente l'évolution de la taille d'un groupe social donnée, chaque barre représente la taille du groupe dans une année donnée (la liste d'années est la colonne à gauche)



**Distribution spatiale du groupe social GS-3  
pour l'année 1992**

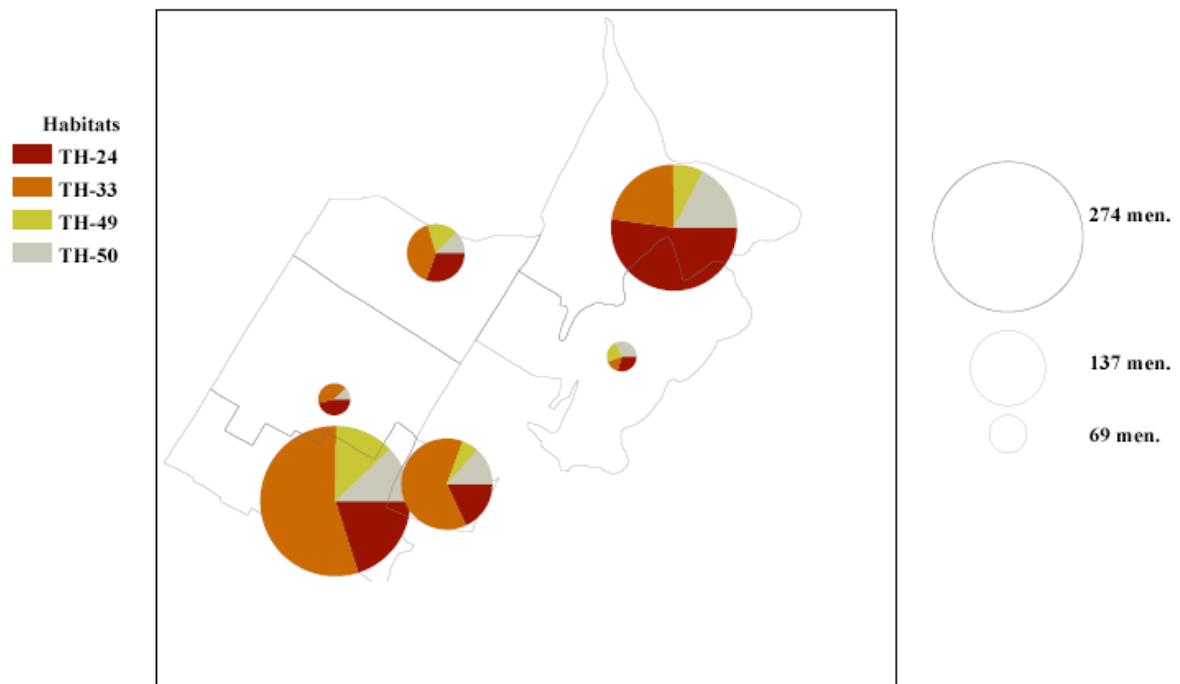


Figure 43. Exemple de carte d'occupation. Carte d'occupation du groupe social simulé *GS-3* pour l'année de simulation 1992, les secteurs urbains graphiques constituent l'arrondissement « La Candelaria » à Bogotá.

Figure 44. Chapitre VIII : Algorithmes de formation automatique de groupes

## **Chapitre VIII : Algorithmes de formation automatique de groupes**

Dans cette partie nous présentons quatre algorithmes de classification implémentés dans la plateforme de simulation que nous avons développée. Le choix des algorithmes implémentés se base sur les conditions suivantes :

- Il s'agit d'algorithmes totalement automatiques, notamment quant à la détection automatique du nombre de classes.
- Ils doivent s'intégrer facilement dans la plateforme de simulation développée.
- Ils doivent prendre en compte l'évolution des classes.

Le premier de ces algorithmes est un algorithme de classification dite *basique* qui produit des classes avec des données (profils de ménages ou profils de logements) identiques. Ce premier algorithme est en effet un algorithme utilitaire, qui sert à réduire l'espace de données sur lequel les trois autres algorithmes effectuent des classifications. Le deuxième algorithme est une version améliorée des cartes auto-organisatives de Kohonen (cf. section 3.1.1.1.2.1.). Le troisième algorithme est une version améliorée de l'algorithme K-means (cf. section 3.1.1.1.1.). Le quatrième est un algorithme basé sur les nuages de particules (cf. section 3.1.2.3.) que nous avons proposé initialement dans [Gil-Quijano & Piron, 2007]. L'algorithme K-means, comme nous l'avons vu précédemment, ne détermine pas de manière automatique le nombre de classes. Nous l'avons donc amélioré pour choisir la « meilleure » partition parmi un certain nombre de classes produites par l'algorithme classique. L'algorithme de cartes auto-organisatives de Kohonen détermine (en fonction du paramétrage et notamment de la configuration de la grille) le nombre de classes d'une partition de manière automatique ; toutefois, les résultats dépendent fortement des poids synaptiques initiaux (normalement choisis de manière aléatoire). Nous avons donc introduit une amélioration similaire à celle introduite dans l'algorithme K-means pour choisir la « meilleure » partition parmi plusieurs produites par l'algorithme classique de Kohonen.

Finalement, l'algorithme basé sur les nuages de particules a pour objectif principal d'augmenter le déterminisme dans les résultats de la classification. De cette manière, il utilise une approche globale et déterministe de l'initialisation et de l'actualisation des positions des

particules, au lieu de l'approche localisée et stochastique typique des algorithmes « classiques » basés sur les nuages de particules (cf. 3.1.2.3.). D'autre part, cet algorithme s'adapte mieux que les algorithmes K-means et de Kohonen dans la prise en compte de l'évolution des groupes.

Ces différents algorithmes sont implémentés par les classes dans le *package* « Formation de Groupes ». Il s'agit d'implémentations de la classe abstraite *ConstructeurClusters*. Ces implémentations sont présentées dans le diagramme ci-dessous.

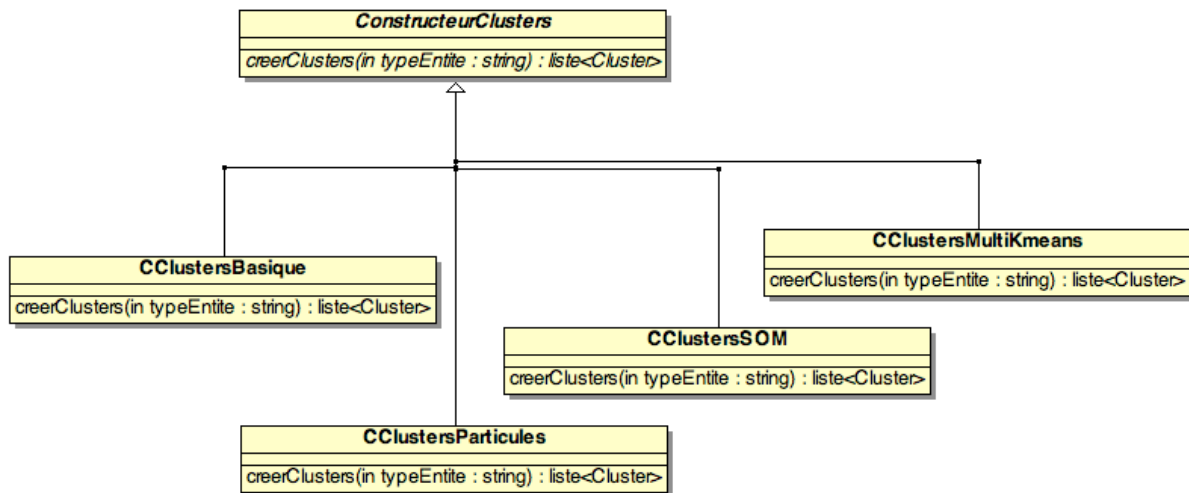


Figure 45. Diagramme de classe. Différents implémentations de la classe *ConstructeurClusters*

Les algorithmes de classification automatique sont implémentés dans la méthode *creerClusters* des différentes classes. Dans cette méthode, le paramètre d'entrée « *typeEntite* » définit sur quelles micro-entités est effectué la classification automatique. Il peut donc prendre les valeurs : « ménages » ou « logements ».

## 8.1. Classification automatique basique

La classification automatique basique implémente un algorithme simple qui permet de créer des classes composées de micro-entités identiques : chaque classe ne contient que des micro-entités qui ont exactement le même profil. L'algorithme est implémenté dans la classe *CclustersBasique*. Cet algorithme a principalement deux utilités :

- Il est utilisé pour créer les matrices de distribution initiale et finale utilisées dans la construction des règles d'évolution de la population microscopique ;

- Il est utilisé par les autres algorithmes de classification automatique implémentés pour réduire le nombre de micro-entités à traiter : les micro-entités identiques sont représentées par une seule micro-entité avec un poids égal au nombre de micro-entités représentées.

## 8.2. Adaptation des cartes auto-organisatives

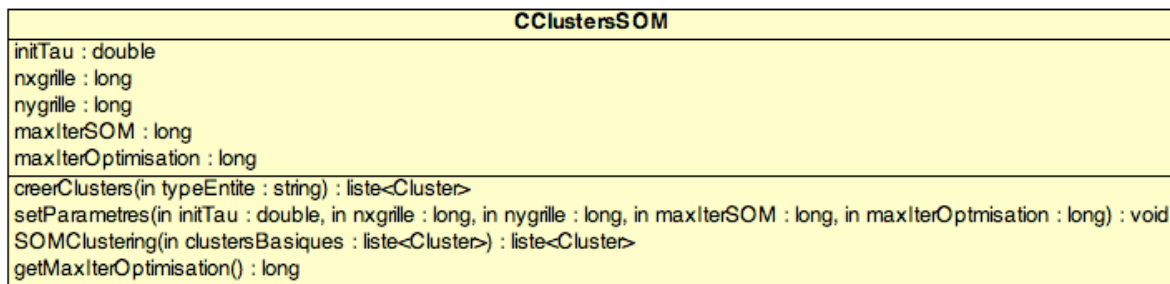


Figure 46. Diagramme de classe. La classe *CClustersSOM*

Dans l'algorithme de classification automatique avec des cartes auto-organisatives présenté dans l'état de l'art (cf. 3.1.1.1.2.1.), les classes résultantes dépendent fortement des poids synaptiques initiaux. Dans cet algorithme-là, les poids des neurones sont initialisés de manière aléatoire. En raison de ces caractéristiques, pour un même ensemble de données, différentes exécutions de l'algorithme produisent des résultats différents. Pour palier cette limitation, nous avons introduit une optimisation qui permet de choisir la « meilleure » partition parmi les partitions produites par plusieurs exécutions de l'algorithme sur le même ensemble de données. Le choix est fait par la comparaison de l'indice de Davis-Bouldin de chaque partition (cf. 3.1.3.1.2.). Le meilleur ensemble est celui qui a la valeur de Davis-Bouldin minimale. L'algorithme optimisé est implémenté dans la méthode *creerClusters* de la classe *CClustersSOM* (voir diagramme de classe ci-dessus). Le déroulement de l'algorithme est présenté sous la forme du diagramme de séquence ci-après. L'algorithme classique de cartes auto-organisatives est implémenté dans la méthode *SOMClustering*. Dans cette méthode, nous utilisons des cartes rectangulaires dont les dimensions sont définies par les attributs *nxgrille* et *nygrille*. L'attribut *initTau* définit la valeur initiale du paramètre  $\tau$  d'actualisation des poids des neurones (cf. 3.1.1.1.2.1.). L'attribut *maxIterSOM* définit le nombre d'actualisations des poids des neurones. La mesure de distance utilisée dans cet algorithme est la distance euclidienne entre les poids des neurones et les vecteurs des

caractéristiques des micro-entités exprimés sous forme disjonctive complète (cf. Annexe III, Calcul des distances entre profils). Pour faire évoluer la valeur de  $\tau$  nous utilisons la formule suivante :

$$\tau(t) = \text{initTau} \times \left(1 - \frac{t}{\text{maxIterSOM}}\right)$$

Où  $t$  est le nombre courant d'actualisations des poids de neurones effectués.

De cette manière  $\tau$  est initialement égal à  $\text{initTau}$  ( $t = 0$ ) et descend linéairement jusqu'à atteindre sa valeur minimale dans la dernière actualisation ( $t = \text{maxIterSOM} - 1$ ).

Le voisinage autour du neurone vainqueur est défini par le cercle centre dans le neurone vainqueur, dont le rayon est donné par l'expression suivante :

$$\text{rayon}(t) = \text{maxRayon} \times \left(1 - \frac{t}{\text{maxIterSOM}}\right)$$

Où :

$$\text{maxRayon} = \sqrt{(\text{nxgrille}^2 + \text{nygrille}^2)}$$

On considère qu'un neurone  $i$  est dans le voisinage du neurone vainqueur si :

$$\sqrt{((x_i - x_{\text{vainqueur}})^2 + (y_i - y_{\text{vainqueur}})^2)} < \text{rayon}(t)$$

Où  $x_i$  et  $y_i$  sont les coordonnées  $x$  et  $y$  du neurone  $i$  dans la grille. Autrement dit on considère qu'un neurone est dans le voisinage du vainqueur, si sa distance euclidienne au vainqueur dans la grille est inférieure au rayon. La fonction de voisinage pour le neurone  $i$  à l'itération  $t$  est alors définie par l'expression suivante :

$$\Theta(i, t) = \begin{cases} 1 & \text{si } \sqrt{((x_i - x_{\text{vainqueur}})^2 + (y_i - y_{\text{vainqueur}})^2)} < \text{rayon}(t) \\ 0 & \text{si } \sqrt{((x_i - x_{\text{vainqueur}})^2 + (y_i - y_{\text{vainqueur}})^2)} \geq \text{rayon}(t) \end{cases}$$

De cette manière, le voisinage inclut au début toutes les neurones dans la grille ( $\text{rayon} = \text{maxRayon}$  pour  $t = 0$ ) et se rétrécit de manière linéaire, pour atteindre sa valeur minimale dans la dernière actualisation des poids ( $t = \text{maxIterSOM} - 1$ ).

Il faut noter que le nombre de classes, en plus d'être dépendant des valeurs initiales des poids synaptiques, est aussi dépendant du nombre de neurones dans la grille. Pour l'instant, le choix des dimensions de la grille et des paramètres  $\text{initTau}$  et  $\text{maxIterSOM}$  sont laissés à l'utilisateur du système.

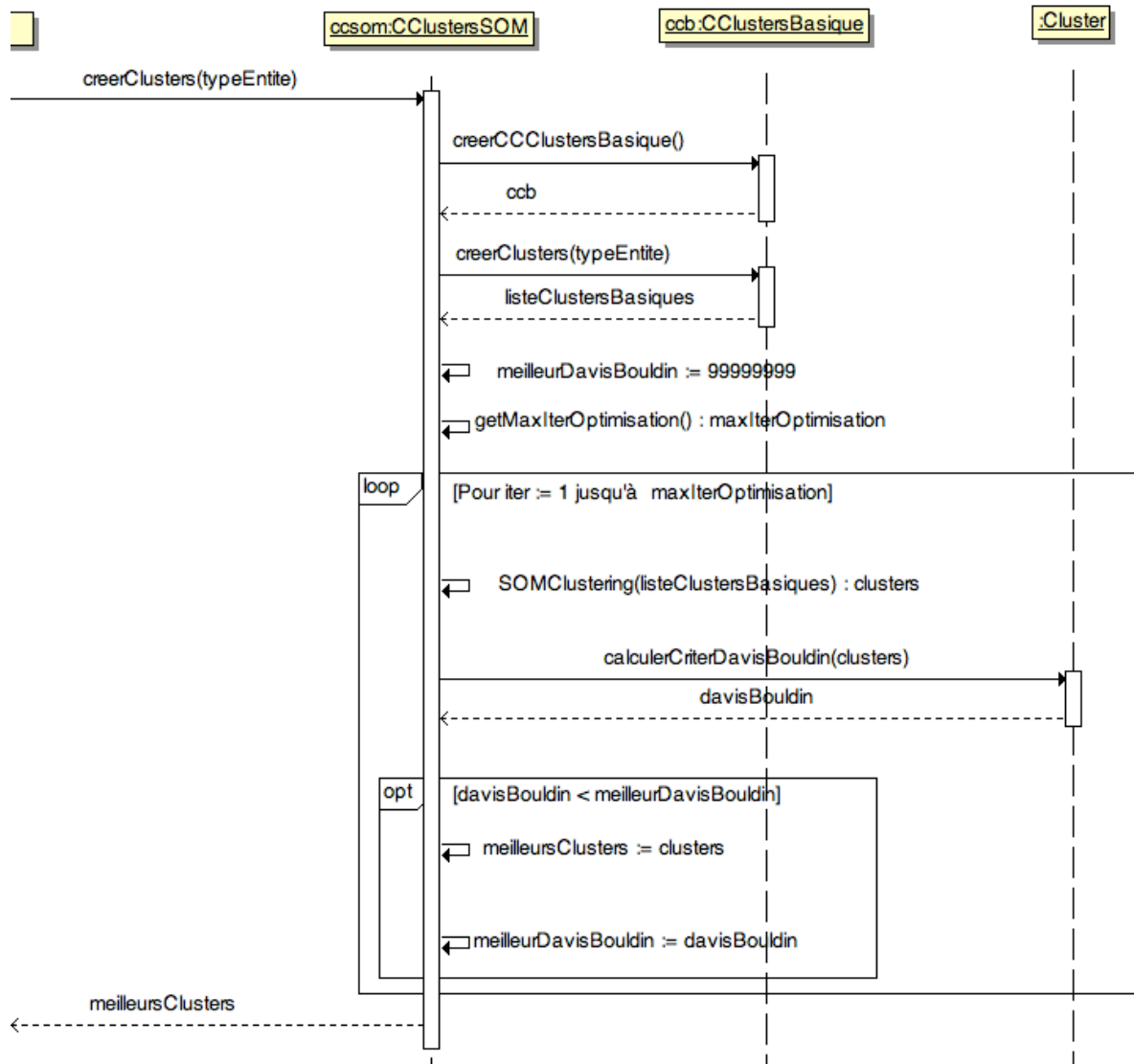


Figure 47. Diagramme de séquence. Optimisation de la classification automatique avec une carte auto-organisative de Kohonen

### 8.3. Multi-Kmeans

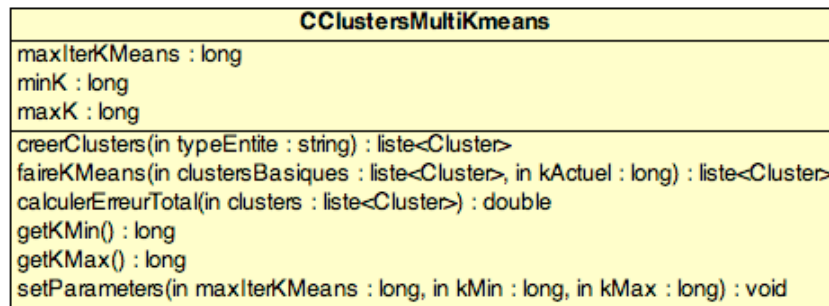


Figure 48. Diagramme de classe. La classe *CClustersMultiKmeans*

L'algorithme K-Means présente deux difficultés : le nombre de classes doit être donné à l'avance et le résultat est dépendant des centres initiaux (généralement choisis de manière aléatoire). La dépendance par rapport aux centres initiaux est résolue dans l'algorithme classique par une sélection de la partition qui produit l'erreur minimale. Il reste à résoudre le problème du choix optimal du nombre de classes. Nous proposons une solution basé sur le calcul de l'indice de Davis-Bouldin (cf. 3.1.3.1.2.). L'idée est de faire varier le nombre de classes et de choisir la « meilleure » partition (celle qui minimise la valeur de l'indice de Davis-Bouldin) parmi toutes les partitions produites. Nous appelons cet algorithme *Multi-Kmeans*, et nous l'implémentons dans la classe *CClustersMultiKmeans* (voir diagramme de classe ci-dessus). Les attributs *kMin* et *kMax* définissent l'intervalle dans lequel on va faire varier le nombre de classes. L'attribut *maxIterKMeans* définit le nombre de fois que les partitions sont actualisées pour une même valeur du nombre de classes. L'algorithme *Multi-Kmeans* est implémenté dans la méthode *creerClusters* de la classe *CClustersMultiKmeans* (voir diagramme de séquence ci-dessous). L'algorithme classique de *Kmeans* est implémenté dans la méthode *faireKmeans*. Dans cet algorithme, nous utilisons la distance euclidienne entre profils des micro-entités, exprimée sous forme disjonctive complète. (cf. Annexe III, Calculs de distances entre profils).

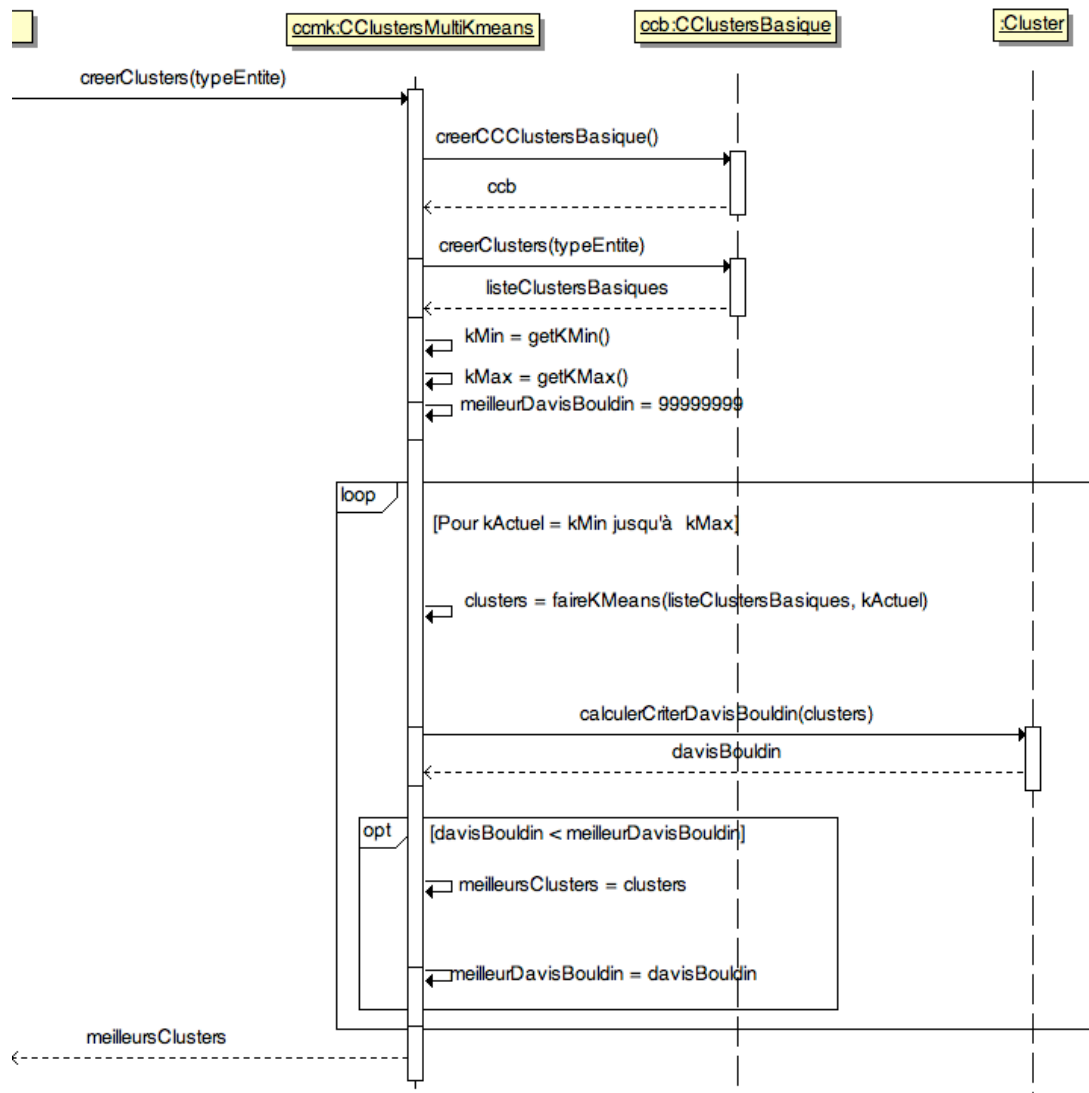


Figure 49. Diagramme de séquence. Algorithme de classification automatique *MultiKMeans*

## 8.4. Algorithme basé sur les nuages de particules

Nous avons développé un algorithme basé sur les nuages de particules (cf. 3.1.2.3.). Cet algorithme a été originalement présenté dans [Gil-Quijano & Piron, 2007]. Les principes pour la construction de cet algorithme sont :

- Réaliser un algorithme automatique avec un nombre de classes déterminé de manière automatique.
- Tenir compte de l'évolution des groupes comme conséquence de l'évolution de la population et du parc de logements (ajout/suppression de ménages/logements)



- Avoir un algorithme le plus déterministe possible.

Dans les algorithmes de classification « classiques » basés sur les nuages de particules, on retrouve souvent les caractéristiques suivantes :

- La localisation initiale des particules est aléatoire.
- L'actualisation de la localisation des particules se fait par rapport à un voisinage et par l'interaction avec les particules dans ce voisinage.
- L'espace d'interaction entre particules est discrétisé souvent sous forme de grille rectangulaire.

La première caractéristique vient du fait que l'on ne veut pas faire des hypothèses sur la localisation initiale des particules : cette mesure a un effet pervers sur la stabilité de l'algorithme. En effet, il est très peu probable de retrouver la même partition pour plusieurs exécutions de l'algorithme sur un même jeu de données. De notre côté, nous proposons un algorithme totalement déterministe d'initialisation des localisations des particules, basé sur la différence entre les données qu'elles représentent. De cette manière la stabilité de l'algorithme est accrue.

Les deuxième et troisième caractéristiques des algorithmes classiques sont prises en compte avec l'objectif d'améliorer la performance en termes de temps calcul. En effet, le nombre de calculs est réduit quand on considère des actualisations locales et quand l'espace est discrétisé. La prise en compte du voisinage et une considération discrétisée de l'espace d'interaction introduisent deux paramètres : la taille du voisinage et la taille de la grille de discrétisation de l'espace. Nous avons pu vérifier expérimentalement que les partitions résultantes sont très dépendantes des valeurs de ces deux paramètres. En général, nous avons pu observer que le nombre de regroupements de particules produit par l'algorithme augmente avec la diminution de la taille du voisinage et de la taille de la grille. Pour éviter la dépendance envers ces deux paramètres, nous proposons l'utilisation d'un espace continu et considérons des interactions avec toutes les particules. Certes, nous perdons en performance en termes de temps de calcul, mais nous gagnons en simplicité et stabilité de l'algorithme.

A partir de ces considérations, nous proposons un modèle d'inspiration physique, où chaque groupe de micro-entités identique est représenté par une particule avec un poids. L'algorithme est divisé en deux étapes fondamentales : la formation des amas de particules et la création des classes basés sur un mécanisme de détection des amas. La formation des amas est basée sur un mécanisme d'attraction/répulsion de particules. La détection des amas est en effet un processus de classification automatique sur l'espace d'interaction où les particules sont localisées.

### 8.4.1. Propriétés des particules

Les particules sont des agents réactifs [Drogoul & Ferber, 1994] qui représentent des micro-entités (ménages ou logements) caractérisés par  $p$  modalités socio-démographiques. Dans le cas de Bogota, nous considérons  $p=43$  modalités pour décrire les ménages et  $p=29$  modalités pour décrire les logements (cf. Annexe I, Description des caractéristiques des ménages et des logements dans la ville de Bogota). Il existe un espace d'interaction dans lequel les agents sont localisés, communiquent et se déplacent par des mécanismes simples d'action/réaction. Ces mécanismes sont définis à partir d'une force d'attraction/répulsion avec les autres agents selon qu'ils se ressemblent ou non. Chaque agent modifie, à chaque itération, sa localisation dans cet espace en fonction de la force d'attraction/répulsion qu'il subit.

#### 8.4.1.1. Différence de comportements entre agents

Les caractéristiques des micro-entités sont mises sous forme disjonctive complète (cf. 3.1.). Pour déterminer la différence entre deux agents  $a$  et  $b$  par rapport à leurs caractéristiques socio-démographiques, nous utilisons la distance de Hamming qui se prête bien au calcul de distance entre un grand nombre de vecteurs binaires [Hamming, 1980] (cf. Annexe III, Calcul des distances entre profils). Cette distance varie dans l'intervalle  $[0,2p]$ . Les agents différents ont une valeur proche de  $2p$  et les agents similaires une valeur proche de  $0$ .

#### 8.4.1.2. Espace d'interaction

Nous considérons comme espace d'interaction, un espace bidimensionnel<sup>11</sup> dans lequel les agents se déplacent. La distance entre deux agents dans cet espace est la distance euclidienne usuelle :

$$D_{Interaction}^2(a, b) = \sum_{i=1}^2 x_i(a) - x_i(b)$$

Où  $x_i(a)$  et  $x_i(b)$  sont les coordonnées des agents  $\mathbf{a}$  et  $\mathbf{b}$  dans cet espace.

Chaque agent évalue la force d'attraction/répulsion qui est exercée sur lui et se déplace dans l'espace d'interaction selon une trajectoire qui dépend de cette force.

#### 8.4.1.3. Force d'attraction/répulsion

---

<sup>11</sup> Nous avons réalisé principalement des tests dans des espaces bidimensionnels afin d'avoir un retour visuel direct du processus de formation de groupes.

La plupart des algorithmes de classification basés sur des interactions entre agents reposent sur une approche d'attraction/répulsion. Beaucoup d'entre eux s'inspirent des animaux sociaux, tel le travail présenté dans [Monmarché et al, 2002] qui adapte la formation de nuages d'insectes à la classification de données ; ou encore celui de [Renault, 2001] qui utilise la diffusion de phéromones pour faire le tri de mails par mot clé. Notre algorithme utilise aussi un mécanisme d'attraction/répulsion, mais l'approche relève davantage de la physique que des animaux sociaux. Ainsi, nous considérons des forces d'attraction/répulsion entre les agents qui peuvent être assimilées aux champs électromagnétiques. Les agents subissent des forces et se déplacent en conséquence. Le calcul de ces forces se base sur la différence entre agents. Chaque agent doit alors trouver l'emplacement optimal, qui minimise la force qu'il subit.

La force  $f(a,b)$  exercée mutuellement entre deux agents **a** et **b** est un vecteur bidimensionnel qui dépend de la localisation des agents (distance d'interaction) et de leur différence de comportements (distance de Hamming). La coordonnée  $i$  de cette force est déterminée de la façon suivante :

$$f_i(a,b) = \text{signe}(a,b) \times \text{AmplForce}(a,b) \times \frac{x_i(a) - x_i(b)}{D_{\text{Interaction}}(a,b)}$$

où  $\text{AmplForce}(a,b)$  désigne l'amplitude de la force  $f(a,b)$  donnée par :

$$\text{AmplForce}(a,b) = \frac{|D_{\text{Hamming}}(a,b) - D_{\text{Interaction}}(a,b)|}{2}$$

et

$$\text{signe}(a,b) = \begin{cases} -1 & \text{si } D_{\text{Hamming}} < D_{\text{Interaction}} \\ 1 & \text{si } D_{\text{Hamming}} \geq D_{\text{Interaction}} \end{cases}$$

Cette force permet de définir la distance d'interaction en fonction de la distance de Hamming. L'objectif est de minimiser cette force, de manière à trouver le bon positionnement des agents (distance d'interaction égale à la distance de Hamming). La distance d'interaction tend donc vers la distance de Hamming et varie entre 0 et  $2p$  (où  $p$  est le nombre de modalités considérées dans la description des micro-entités).

La résultante de la force d'attraction/répulsion que les autres  $n-1$  agents exercent sur un agent **a** donné est la moyenne de toutes les forces exercées sur l'agent (voir ci-dessous).

$$f_{\text{Résultante}_i}(a) = \frac{1}{n-1} \times \sum_{b=1, b \neq a}^n f_i(a,b)$$

L'amplitude de cette force est calculée par :

$$AmplFRésultante(a) = \sqrt{\sum_{i=1}^2 fRésultante_i(a)}$$

#### 8.4.1.4. Energie du système

Pour avoir une mesure globale du rapport entre les différences de comportement des agents (distance de Hamming) et les distances d'interaction, nous calculons la moyenne des forces exercées sur tous les agents. Nous appelons cette mesure « énergie du système », déterminée par :

$$Energie = \frac{1}{n} \times \sum_{a=1}^n |AmplFRésultante(a)|$$

Plus la différence entre les distances d'interaction et les distances de Hamming sont grandes, plus l'énergie du système est grande.

#### 8.4.1.5. Localisation initiale

Partant du principe que nous voulons construire un algorithme le plus déterministe possible, nous avons conçu un mécanisme d'initialisation de la localisation des particules totalement déterministe. Le principe de cet algorithme est de localiser les particules dans l'espace d'interaction par rapport à la différence entre leurs profils. L'algorithme se base sur la construction de « cercles d'interaction » et le calcul d'intersections entre cercles. En effet, si l'on considère deux particules dont la distance de Hamming est  $d$ , on peut tracer un cercle ayant pour centre une des deux particules et comme rayon la valeur de  $d$ . Ce cercle, que l'on appelle *cercle d'interaction* entre les deux particules, définit toutes les localisations possibles dans lesquelles la deuxième particule peut se placer en respectant l'égalité entre la différence de profil et la distance euclidienne sur l'espace d'interaction, c'est-à-dire, toutes les localisations où la force entre les deux particules est égale à 0. Nous appelons la distance  $d$  la *distance à force 0* entre deux particules. Nous appelons *cercle d'interaction de  $P0$  sur  $P1$*  le cercle d'interaction entre les particules  $P0$  et  $P1$  où  $P0$  est localisé dans le centre.

Maintenant, supposons que l'on a déjà localisé deux particules  $P0$  et  $P1$  : les deux particules sont séparées d'une distance  $d_{01}$  égale à leur distance à force 0. Si l'on veut localiser une troisième particule  $P3$  de manière à que les distances entre la nouvelle particule et les particules déjà localisées soient les plus proches des distances à force 0, il suffit de localiser  $P3$  sur une des intersections, (s'il existe une intersection) entre le cercle d'interaction de  $P1$  sur  $P3$  et le cercle d'interaction de  $P0$  sur  $P3$ . Si les deux cercles ne se croisent pas, on peut

localiser la particule  $P3$  dans un point intermédiaire entre les localisations de  $P0$  et  $P1$  qui est déterminé en fonction des distances à force 0 entre  $P3$  et les deux autres particules (cf. figure ci-après).

Dans la détermination du point où l'on peut localiser  $P3$ , il faut tenir compte de quatre cas différents qui sont illustrés dans la figure ci-dessous:

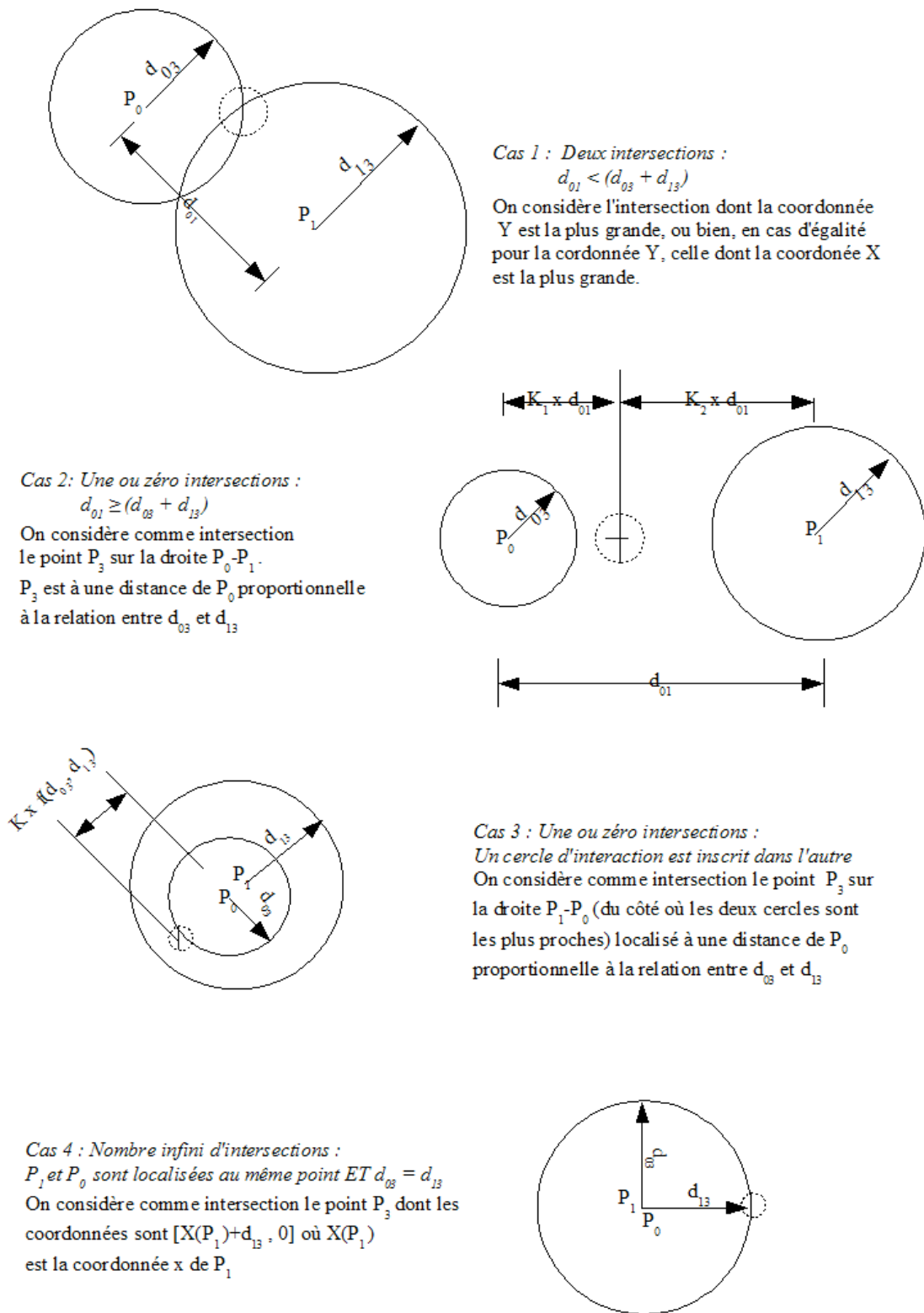


Figure 50. Détermination d'UNE intersection entre deux cercles d'interaction

- Dans le cas général (cas 1) où les deux cercles se croisent en deux points, il faut choisir un des deux points selon une règle déterministe.

- Dans le cas où les cercles sont trop éloignés (cas 2) et ne se croisent pas ou se croisent en un seul point, il faut trouver un point intermédiaire localisé entre les deux cercles.
- Dans le cas où un cercle est inscrit dans l'autre (cas 3), les cercles ne se croisent pas ou se croisent en un seul point et il faut déterminer un point intermédiaire localisé entre les deux cercles.
- Dans le cas où les deux cercles ont le même centre et la même valeur de rayon, il existe un nombre infini d'intersections, il faut donc choisir une des intersections selon une règle déterministe.

Sur la base de ces principes géométriques, nous proposons l'algorithme présenté ci-après pour déterminer la localisation initiale de l'ensemble de particules.

Entrée : ensemble de particules

Sortie : l'ensemble de particules avec leur localisation initialisée

DEBUT

1. Insérer toutes les particules dans la liste *particules non considérées*
2. Créer la liste vide *particules déjà considérées*
3. Déterminer le couple de particules  $[P0 ; P1]$  dont la distance de Hamming  $d$  est la plus petite de toutes les distances entre toutes les couples possibles.
4. Localiser la particule  $P0$  dans les coordonnées  $(0, 0)$
5. Localiser la deuxième particule du couple dans les coordonnées  $(0, d)$
6. Eliminer les particules  $P0$  et  $P1$  de la liste *particules non considérées*
7. Créer une particule moyenne  $PC$  dont le profil est le profil moyen entre les profils de  $P0$  et  $P1$
8. TANT QUE il y a des particules dans la liste *particules non considérées*

FAIRE

- a. Choisir la particule  $PP$  dont la distance de Hamming à la particule moyenne  $PC$  est la plus petite.
- b. Créer une liste vide d'*intersections*
- c. *taille\_considerées* \_ nombre de micro-entités dans la liste *particules déjà considérées*
- d. POUR  $i$  de 1 à  $(taille\_considerées - 1)$

FAIRE

- $PI\_i$  particule à la position  $i$  dans la liste de *particules déjà considérées*
- $PII\_i$  particule à la position  $i+1$  dans la liste de *particules déjà considérées*
- Déterminer UNE *intersection* (voir figure ci-dessus) entre les cercles d'interaction de  $PI$  sur  $PP$  et de  $PII$  sur  $PP$ .
- Ajouter l'*intesection* à la liste d'*intersections*

FIN\_POUR

- e. Déterminer l'*intersection moyenne* à partir des intersections contenues dans la liste d'*intersections*
- f. Localiser la particule  $PP$  dans l'*intersection moyenne*

g. Éliminer la particule *PP* de la liste de *particules non considérées*  
 h. Ajouter la particule *PP* à la fin de la liste de *particules déjà considérées*  
 FIN\_TANT\_QUE  
 9. renvoyer *liste de particules déjà considérées*  
 FIN

Encadré 11. Algorithme d'initialisation de la localisation des particules

### 8.4.2. Algorithme de classification des agents

L'algorithme de classification des agents (voir figure ci-dessous) est fondé sur la minimisation de l'énergie du système. Cela équivaut à trouver l'ensemble des localisations qui minimisent la force subie par chaque agent.

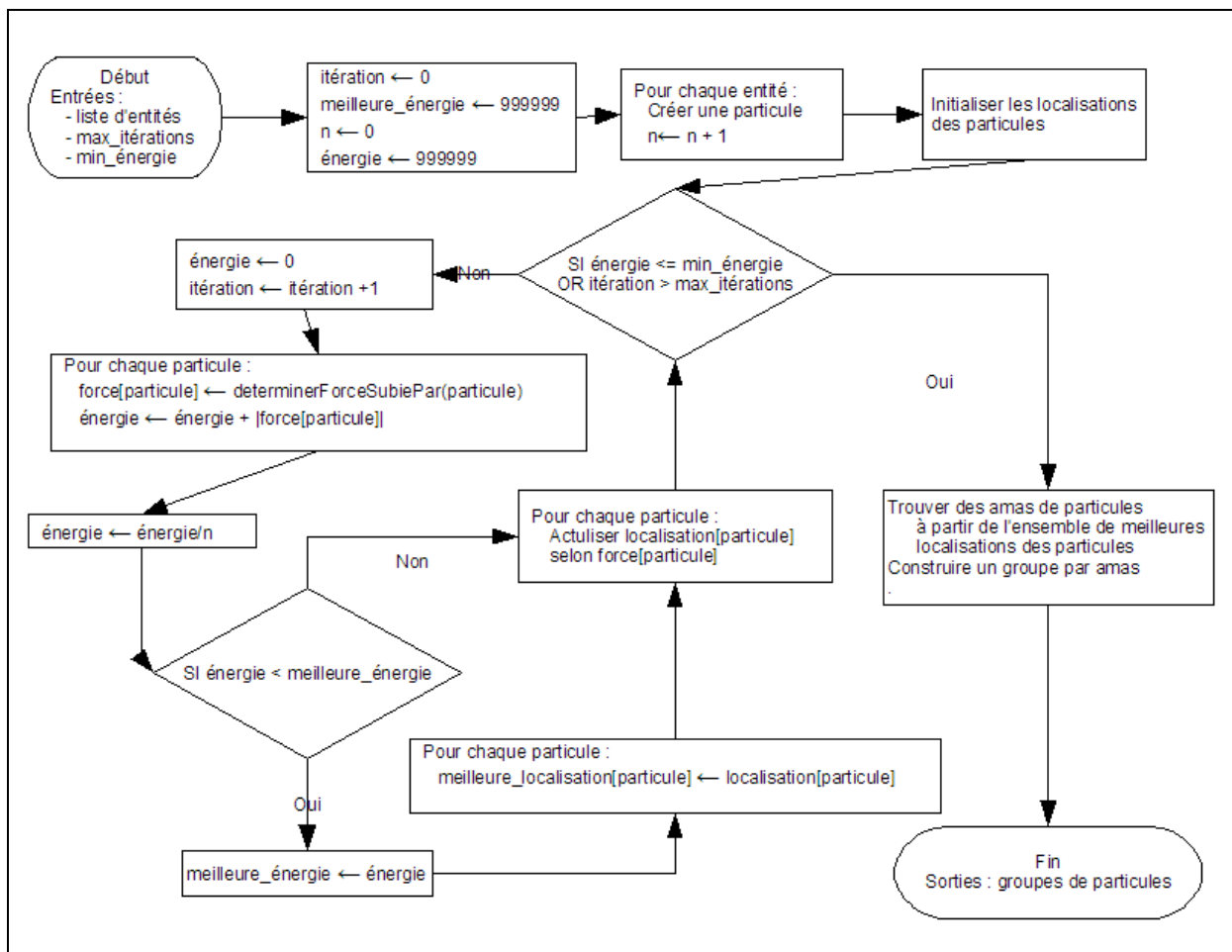


Figure 51. Description générale de l'algorithme de formation de groupes de particules

Cet algorithme propose une approche de type *blackboard* (blackboards systems [Corkill, 1991]) où l'on calcule d'abord les forces subies par chacune des particules à l'instant  $t=i$  et ensuite on déplace chaque particule selon la force subie en  $t=i$  vers sa localisation à l'instant



$t=i+1$ . Ce mécanisme permet de simuler un calcul simultané des prochaines localisations sans avoir recours à plusieurs fils d'exécution.

#### 8.4.2.1. Constitution des groupes

Initialement, les agents se localisent selon l'algorithme d'initialisation dans l'espace d'interaction. Par la suite, chaque agent cherche à se localiser de façon à réduire la force qui est exercée sur lui par les autres agents. À chaque itération, un agent calcule la résultante de la force d'attraction/répulsion que les autres exercent sur lui et se déplace dans la direction de la force résultante d'une distance proportionnelle à l'amplitude  $AmplFRésultante(a)$  de cette force. L'amplitude de ce déplacement est déterminée par l'équation suivante :

$$AmplDéplacement(a) = FacteurAccélération \times AmplFRésultante(a)$$

Dans cette équation,  $Facteur\_Accélération$  est une constante de proportionnalité positive (cf. 8.4.2.3 optimisation de l'algorithme). Puisqu'il s'agit de trouver la configuration (ensemble des localisations) des agents qui minimise l'énergie du système, l'actualisation de la localisation des agents est faite jusqu'à ce que l'énergie ait atteint un seuil minimum ( $min\_énergie$ ) ou bien que le nombre d'itérations ait atteint un maximum ( $max\_itérations$  -cf. figure 47 ci-dessus).

Ces seuils sont définis par l'utilisateur du système. Pour aider dans le choix de ces seuils, nous construisons automatiquement des courbes d'évolution de l'énergie du système. Avec une première phase d'expérimentation et à partir de la courbe d'énergie du système, on peut adapter ces seuils. Un processus de détection de groupes d'agents est alors lancé sur la configuration correspondante à l'énergie minimale.

Afin d'illustrer l'évolution des localisations des particules dans une classification nous présentons dans la figure ci-dessous des images de la distribution de particules dans l'espace d'interaction pour une classification réalisée sur des données de ménages pour l'année 1973 dans l'arrondissement « La Candelaria » à Bogotá. Afin de comparer, nous présentons l'évolution de la distribution des particules avec deux initialisations différentes :

- initialisation aléatoire : les particules sont placées aléatoirement dans une région rectangulaire de taille  $2p \times 2p$  ( $p$  étant le nombre de modalités considérées et  $2p$  la distance de Hamming maximale possible entre deux particules, cf. section 8.4.1.3).
- initialisation avec l'algorithme basé sur les intersections entre cercles d'interaction présenté dans la section précédente.

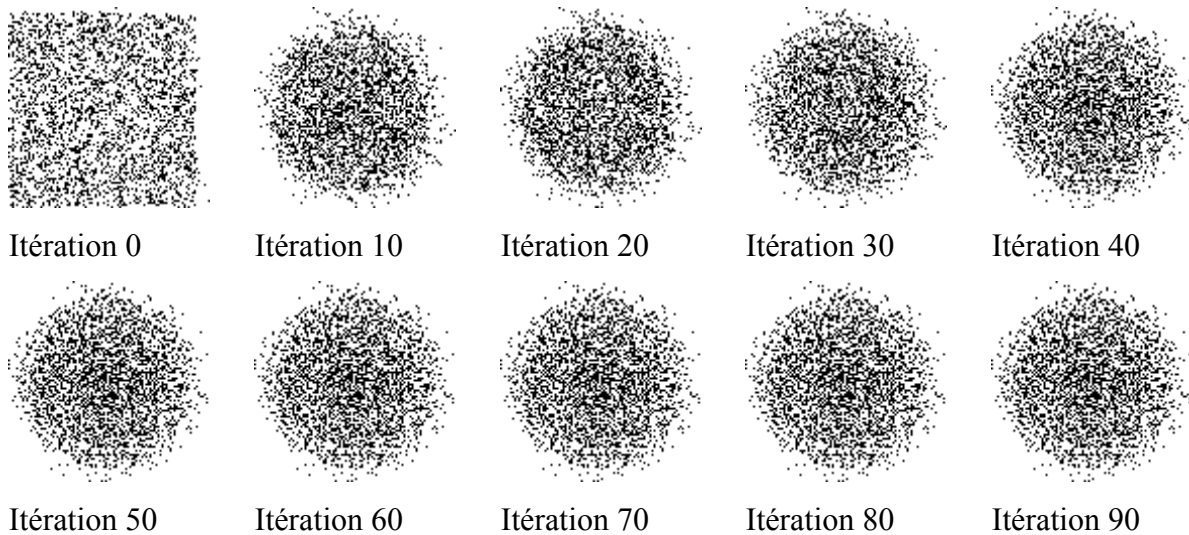


Figure 52. Evolution des localisations des particules à partir d'une distribution initiale aléatoire. Classification faite sur des données des ménages de l'arrondissement « La Candelaria » à Bogotá, pour le recensement de 1973.

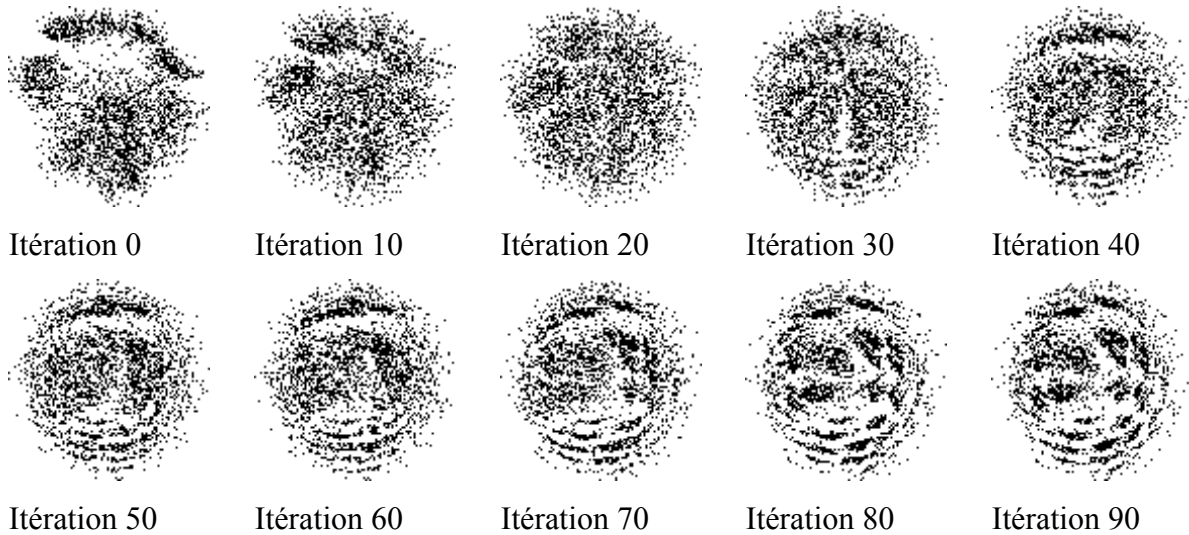


Figure 53. Evolution des localisations des particules à partir d'une distribution initiale construite à partir de l'algorithme basé sur les intersections de cercles d'interaction. Classification faite sur des données des ménages de l'arrondissement « La Candelaria » à Bogotá, pour le recensement de 1973.

On peut observer clairement dans les figures présentées que la distribution des particules dans l'espace d'interaction évolue de manière plus rapide vers une distribution « structurée » lorsque l'initialisation est faite par l'algorithme basé sur les intersections de cercles d'interaction.

D'ailleurs, avec l'initialisation aléatoire, on tombe dans un minimum local de l'énergie du système et l'évolution de la distribution des particules dans l'espace d'interaction stagne à partir de l'itération 50.

#### 8.4.2.2. Détection des groupes

La contrainte de cet algorithme est de ne pas fixer le nombre de groupes. La plupart des algorithmes qui déterminent le nombre de classes de façon automatique sont des algorithmes hiérarchiques tel que l'algorithme X-means [Pelleg & Moore, 2000] : la « meilleure » partition est choisie selon un indice de sa qualité notamment le BIC (Bayesian Information Criterion), le AIC (Akaike Information Criteria), R-Square, ou CCC (Cubic Clustering Criterion) [Nakache et Confais, 2003]. Dans une autre optique, les algorithmes de classification basés sur les systèmes multi-agents et en particulier ceux basés sur les nuages de particules, n'ont pas besoin de définir le nombre de classes [Azzag et al., 2004]. Par contre, ils introduisent souvent d'autres paramètres tels que des seuils de similarité pour définir l'appartenance d'un individu à un groupe [Monmarché et al, 2002]. Ne voulant pas introduire de nouveaux paramètres, nous nous référons à la distribution des agents dans l'espace d'interaction. Nous utilisons ici l'algorithme Multi-Kmeans avec comme données d'entrée les coordonnées des micro-entités. Comme on l'a présenté précédemment, cet algorithme choisit la « meilleure » partition selon l'indice de Davis-Bouldin.

A manière d'illustration, nous présentons dans la figure ci-dessous, le résultat de la détection de groupes sur la distribution des particules montrée dans la section précédente. Le paramétrage utilisé pour l'algorithme Multi-Kmeans est le suivant :

- Nombre minimale de classes (*minK*): 2
- Nombre maximale de classes (*maxK*): 20
- Nombre d'actualisations des centres pour un même nombre de classes (*maxIterKmeans*): 100.

Avec ce paramétrage l'algorithme trouve 15 classes (voir figure ci-dessous).

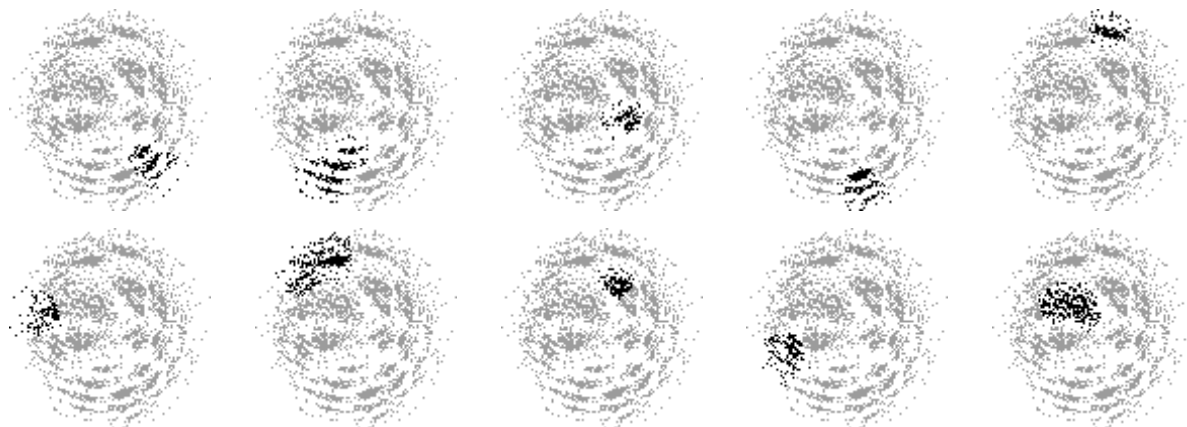




Figure 54. Classes trouvées par l'algorithme de détection de classes sur la distribution des particules dans l'espace d'interaction. Classification faite sur des données des ménages de l'arrondissement « La Candelaria » à Bogotá, pour le recensement de 1973.

#### 8.4.2.3. Optimisation de l'algorithme

Afin d'optimiser l'algorithme, nous introduisons un facteur d'accélération dans le calcul de l'amplitude du déplacement. La valeur de ce facteur d'accélération change en fonction de l'évolution de l'énergie du système. L'objectif de cette optimisation est d'éviter les configurations avec des minima locaux de l'énergie. Comme on peut constater dans l'équation de calcul de l'amplitude du déplacement, on obtient un effet d'amplification de la force pour des valeurs du facteur d'accélération supérieures à 1, tandis que pour des valeurs inférieures à 1, on obtient un effet d'atténuation de la force. L'optimisation consiste à augmenter le facteur d'accélération lorsque la vitesse de diminution de l'énergie du système est petite. On veut ainsi « pousser » les agents vers des configurations qui produisent une diminution d'énergie plus importante. Ce type d'optimisation est proche de la méthode de recuit simulé (Simulated Annealing [Laarhoven & Aarts, 1987]). Pour faire varier le facteur d'accélération nous utilisons deux règles :

- Règle 1 : si l'énergie diminue trop lentement le facteur d'accélération est augmenté. Par défaut, nous augmentons le facteur d'accélération de 10%. Ce pourcentage est réglable par l'utilisateur du système.
- Règle 2 : si le changement de l'énergie est négatif, c'est-à-dire si l'énergie du système augmente, le facteur d'accélération est diminué. Par défaut nous diminuons le facteur d'accélération de 10%. Ce pourcentage est également réglable.

Les pourcentages d'augmentation et de diminution du facteur d'accélération sont choisis après une première étape d'expérimentation, suivant l'évolution de la courbe d'énergie du système. Dans la règle 1, on suppose que la trajectoire d'un agent donné varie très peu d'une itération à l'autre, on peut donc le faire avancer un peu plus vite dans sa trajectoire courante. La règle 2 permet d'éviter que des facteurs d'accélération trop grands produisent des augmentations dans l'énergie du système. Pour mesurer la vitesse de changement de l'énergie, on la compare à la

vitesse de changement dans les itérations initiales. Nous définissons une étape « d'apprentissage » de la vitesse de changement de l'énergie. Dans cette étape, on calcule la moyenne de la vitesse de changement. Nous utilisons par défaut les quatre premières itérations pour déterminer cette vitesse. La règle 1 est alors déclenchée lorsque la vitesse de changement d'énergie est inférieure à 10% de la vitesse dans l'étape d'apprentissage.

Nous avons pu vérifier l'efficacité de cette optimisation [Gil-Quijano et Piron, 2007]. Pour illustrer cette vérification nous présentons dans la figure ci-dessous une comparaison de l'évolution de l'énergie du système pour deux classifications sur les ménages de l'année 1973 de l'arrondissement « La Candelaria » à Bogotá. La première classification a été effectuée avec l'algorithme de nuages de particules sans l'optimisation présentée dans cette section, la deuxième classification prend en compte l'optimisation présentée dans cette section. On observe clairement que dans le cas de l'algorithme optimisé, on atteint des niveaux plus bas d'énergie du système.

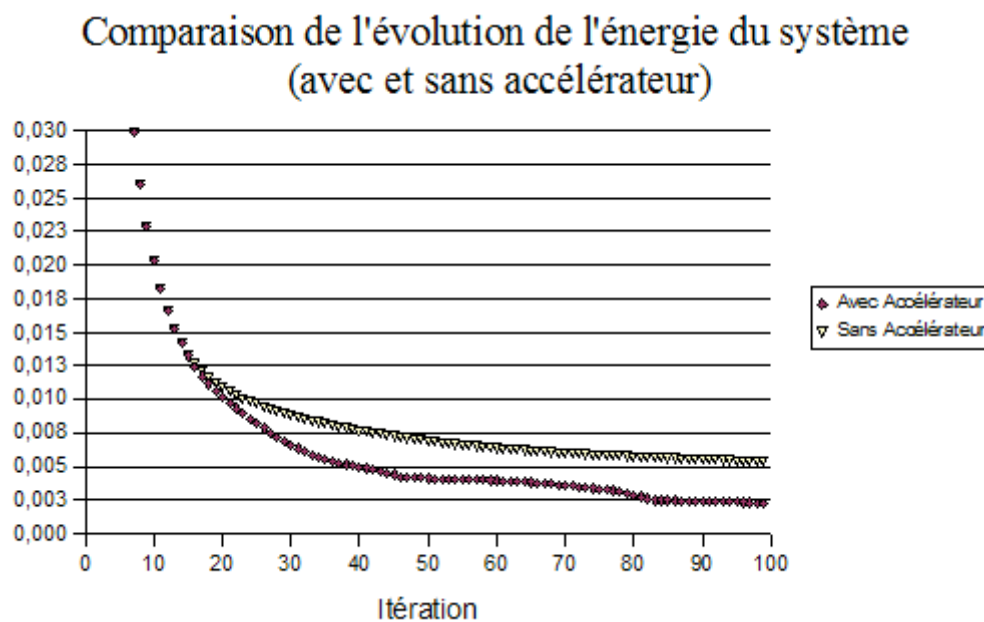


Figure 55. Evolution de l'énergie du système avec et sans accélérateur. Classification faite sur des données des ménages de l'arrondissement « La Candelaria » à Bogotá, pour le recensement de 1973.

#### 8.4.2.4. Prise en compte de l'évolution de l'ensemble de micro-entités

Le processus de formation des amas de particules est basé sur l'optimisation de l'ensemble des localisations des particules. Cette optimisation tient compte des interactions de chaque particule avec toutes les autres. La localisation de chaque particule dépend surtout de l'ensemble des localisations et non pas de la localisation d'une autre particule en particulier. D'autre part, dans le mécanisme d'évolution de la population de ménages et du parc de

logements, on ajoute et on supprime un certain nombre de ménages et de logements d'un certain profil. Partant de la supposition que les changements d'une année sur autre sont très petits par rapport à la population et au parc de logement, nous pouvons proposer une optimisation additionnelle de l'algorithme de classification qui tienne compte de ces changements relativement petits.

Il s'agit simplement de garder les localisations des particules trouvées dans l'année  $i$  et de les utiliser comme localisations initiales de l'année  $i+1$ . De cette manière, on crée un lien entre la localisation de la particule et le profil qu'elle représente. Il faut seulement considérer deux cas particuliers : les profils disparus et les nouveaux profils. Dans le premier cas, tout simplement, on supprime les particules qui représentaient ces profils. Dans le deuxième cas, de nouvelles particules sont ajoutées dans des localisations définies par l'algorithme d'initialisation des localisations.

Ce mécanisme de prise en compte de l'évolution de la population des ménages et du parc de logement présente un avantage principal : le nombre d'actualisations nécessaires pour atteindre les localisations optimales est réduit d'une année à l'autre. De toute manière, dans le cas où les changements dans l'ensemble de micro-entités sont trop importants, la performance dans la qualité des partitions obtenues n'est pas compromise, car dans ce cas-là, on se retrouve tout simplement dans la situation initiale qui nécessite un nombre important d'actualisations pour atteindre les localisations optimales.

## Chapitre IX : Mécanisme de validation

Nous avons implémenté un mécanisme de validation basé sur la comparaison de la matrice de la distribution finale simulée avec la matrice de la distribution finale réelle. Dans le cas pratique de Bogotá, nous ne disposons que des deux recensements de la population (1973 et 1993). Pour cette raison, il n'est pas possible de réaliser des validations de la distribution de population à des années intermédiaires.

Nous effectuons deux types de validation : une validation visuelle et un calcul de la capacité à reproduire la matrice de la distribution de la population finale réelle. La validation visuelle consiste d'une part, en la construction automatique de cartes de comparaison des matrices des distributions des populations finales simulée et de réelle, et d'autre part en la construction d'une image globale de la comparaison de ces deux matrices. Le deuxième type de validation se fait par le calcul d'un coefficient de corrélation entre la matrice de distribution finale simulée et la matrice de distribution finale réelle. La phase de validation est la dernière phase dans une simulation, elle est soulignée en vert dans le macro-algorithme de simulation présenté ci-dessous :

Simulation (*population initiale réelle, population finale réelle, nombre d'années, paramètres additionnels*)

DEBUT

1. Initialiser les paramètres d'évolution de la population
2. Initialiser la *population simulée* à partir de la *population initiale réelle*.
3. POUR *année=1* jusqu'au *nombre d'années*

FAIRE

- a. Evoluer la *population simulée*.
- b. Actualiser la répartition de la *population simulée*.
- c. *année* \_ *année* + 1

FIN POUR

4. Valider la *population simulée* contre la *population finale réelle* :
  - a. Calculer la corrélation entre la *matrice de distribution simulée* et la *matrice de distribution finale réelle*.
  - b. Produire des cartes de validation et d'autres résultats graphiques

FIN

Encadré 12. Phase de validation de la distribution de la population simulée par rapport à la distribution finale réelle dans le macro-algorithme principal de simulation.

## 9.1. Calcul de la corrélation

Pour comparer les matrices de distribution des populations finales simulée et réelle, nous utilisons le coefficient de corrélation croisée ZNCC (Zero mean Normalized Cross-Correlation, cf. Annexe IV, Mesures de corrélation entre deux ensembles de données) qui donne une valeur de corrélation normalisée. Une valeur du ZNCC proche de 0 veut dire que les deux ensembles sont très peu corrélés. Une valeur proche de 1 veut dire que les deux ensembles de données sont fortement corrélés.

Les ensembles de données à comparer sont extraits des matrices de distribution simulée et réelle. En effet, on compare un par un les nombres simulés et réels d'effectifs de logements de chaque type d'habitat occupé dans chaque secteur urbain par chaque groupe social. On compare aussi le nombre de ménages-sans-logement de chaque groupe social. L'algorithme de calcul du coefficient ZNCC est présenté ci-dessous :

```
Caculer_ZNCC (distribution_réelle : matrice de distribution de la population ; distribution_simulée :  
matrice de distribution de la population)
```

```
DEBUT
```

```
14. groupes-sociaux_obtenir_liste_conjointe_groupes_sociaux (distribution_réelle,  
distribution_simulée) // groupes-sociaux est une liste qui contient les groupes sociaux des  
deux matrices de distribution même ceux qui ne sont présents que dans une des matrices.
```

```
15. habitats _ obtenir_liste_conjointe_types_d'habitats(distribution_réelle,  
distribution_simulée)// habitats est une liste qui contient les types d'habitats des deux  
matrices de distribution même ceux qui ne sont présents que dans une des matrices.
```

```
16. secteurs_urbains_ obtenir_liste_conjointesecteur_urbains(distribution_réelle,  
distribution_simulée) // secteurs_urbains est une liste qui contient les secteurs urbains  
représentés par les deux matrices de distribution même ceux qui ne sont présents que dans  
une des matrices.
```

```
//ON CALCULE D'ABORD LES MOYENNES DES DEUX ENSEMBLES DE DONNEES
```

```
17. moyenne_réelle_0
```

```
18. moyenne_simulée_0
```

```
19. n_0
```

```
20. POUR CHAQUE groupe_social dans groupes_sociaux
```



FAIRE

a. POUR CHAQUE *secteur\_urbain* dans *secteurs\_urbains*

FAIRE

i. POUR CHAQUE *type\_habitat* dans *habitats*

FAIRE

- *nombre\_réel* \_ nombre de logements du *type\_habitat* dans *secteur\_urbain* occupés par *groupe\_social* dans la matrice *distribution\_réelle*
- *nombre\_simulé* \_ nombre de logements du *type\_habitat* dans *secteur\_urbain* occupés par *groupe\_social* dans la matrice *distribution\_simulée*
- *moyenne\_réelle* \_ *moyenne\_réelle* + *nombre\_réel*
- *moyenne\_simulée* \_ *moyenne\_simulée* + *nombre\_simulé*
- *n\_n* +1

FIN\_POUR

FIN\_POUR

b. *nombre\_réel* \_ nombre de ménages-sans-logement du *groupe\_social* dans la matrice *distribution\_réelle*

c. *nombre\_simulé* \_ nombre de ménages-sans-logement du *groupe\_social* dans la matrice *distribution\_simulée*

d. *moyenne\_réelle* \_ *moyenne\_réelle* + *nombre\_réel*

e. *moyenne\_simulée* \_ *moyenne\_simulée* + *nombre\_simulé*

f. *n\_n* +1

FIN\_POUR

21. *moyenne\_réelle* \_ *moyenne\_réelle* / *n*

22. *moyenne\_réelle* \_ *moyenne\_réelle* / *n*

//ON CALCULE LE COEFFICIENT

23. *produitScalaire\_0*

24. *sumQuarrésDiffSimulées\_0*

25. *sumQuarrésDiffRéelles\_0*

26. POUR CHAQUE *groupe\_social* dans *groupes\_sociaux*

FAIRE

a. POUR CHAQUE *secteur\_urbain* dans *secteurs\_urbains*

FAIRE

i. POUR CHAQUE *type\_habitat* dans *habitats*

FAIRE

- *nombre\_réel* \_ nombre de logements du *type\_habitat* dans *secteur\_urbain* occupés par *groupe\_social* dans la matrice *distribution\_réelle*

- $\text{nombre\_simulé} \_ \text{nombre de logements du } \text{type\_habitat} \text{ dans } \text{secteur\_urbain}$  occupés par  $\text{groupe\_social}$  dans la matrice  $\text{distribution\_simulée}$
- $\text{diff\_réelle} \_ \text{nombre\_réel} - \text{moyenne\_réelle}$
- $\text{diff\_simulée} \_ \text{nombre\_simulé} - \text{moyenne\_simulée}$
- $\text{sumQuarrésDiffSimulées} \_ \text{sumQuarrésDiffSimulées} + (\text{diff\_simulée} * \text{diff\_simulée})$
- $\text{sumQuarrésDiffRéelles} \_ \text{sumQuarrésDiffRéelles} + (\text{diff\_réelle} * \text{diff\_réelle})$
- $\text{produitScalaire} \_ \text{diff\_réelle} * \text{diff\_simulée}$

FIN\_POUR

FIN\_POUR

- b.  $\text{nombre\_réel} \_ \text{nombre de ménages-sans-logement du } \text{groupe\_social}$  dans la matrice  $\text{distribution\_réelle}$
- c.  $\text{nombre\_simulé} \_ \text{nombre de ménages-sans-logement du } \text{groupe\_social}$  dans la matrice  $\text{distribution\_simulée}$
- d.  $\text{diff\_réelle} \_ \text{nombre\_réel} - \text{moyenne\_réelle}$
- e.  $\text{diff\_simulée} \_ \text{nombre\_simulé} - \text{moyenne\_simulée}$
- f.  $\text{sumQuarrésDiffSimulées} \_ \text{sumQuarrésDiffSimulées} + (\text{diff\_simulée} * \text{diff\_simulée})$
- g.  $\text{sumQuarrésDiffRéelles} \_ \text{sumQuarrésDiffRéelles} + (\text{diff\_réelle} * \text{diff\_réelle})$
- h.  $\text{produitScalaire} \_ \text{diff\_réelle} * \text{diff\_simulée}$

FIN\_POUR

27.  $\text{div} \_ \text{racine\_carré} (\text{sumQuarrésDiffRéelles}) * \text{racine\_carré} (\text{sumQuarrésDiffSimulées})$

28. SI  $\text{div}$  est différent de 0

ALORS

a.  $\text{zncc} \_ \text{produitScalaire} / \text{div}$

SINON // Le coefficient n'est pas défini

a.  $\text{zncc} \_ 0$

FIN\_SI

29. renvoyer  $\text{zncc}$

FIN

Encadré 13. Algorithme de calcul de la corrélation entre la population réelle et la population simulée

## 9.2. Validation visuelle

Nous construisons de manière automatique des cartes de validation de la distribution de chaque groupe social dans les secteurs urbains considérés. Un exemple est présenté ci-dessous. Sur la carte de chaque secteur urbain considéré, nous construisons des demi-camemberts qui représentent les différents types d'habitat occupés par chaque groupe social dans le secteur urbain donné. Le demi-camembert supérieur représente le groupe social simulé, le demi-camembert inférieur représente le groupe social réel. La taille des camemberts est proportionnelle au nombre de ménages dans le secteur urbain faisant partie du groupe social représenté. Le mécanisme de construction automatique de ces cartes est décrit dans la partie III.

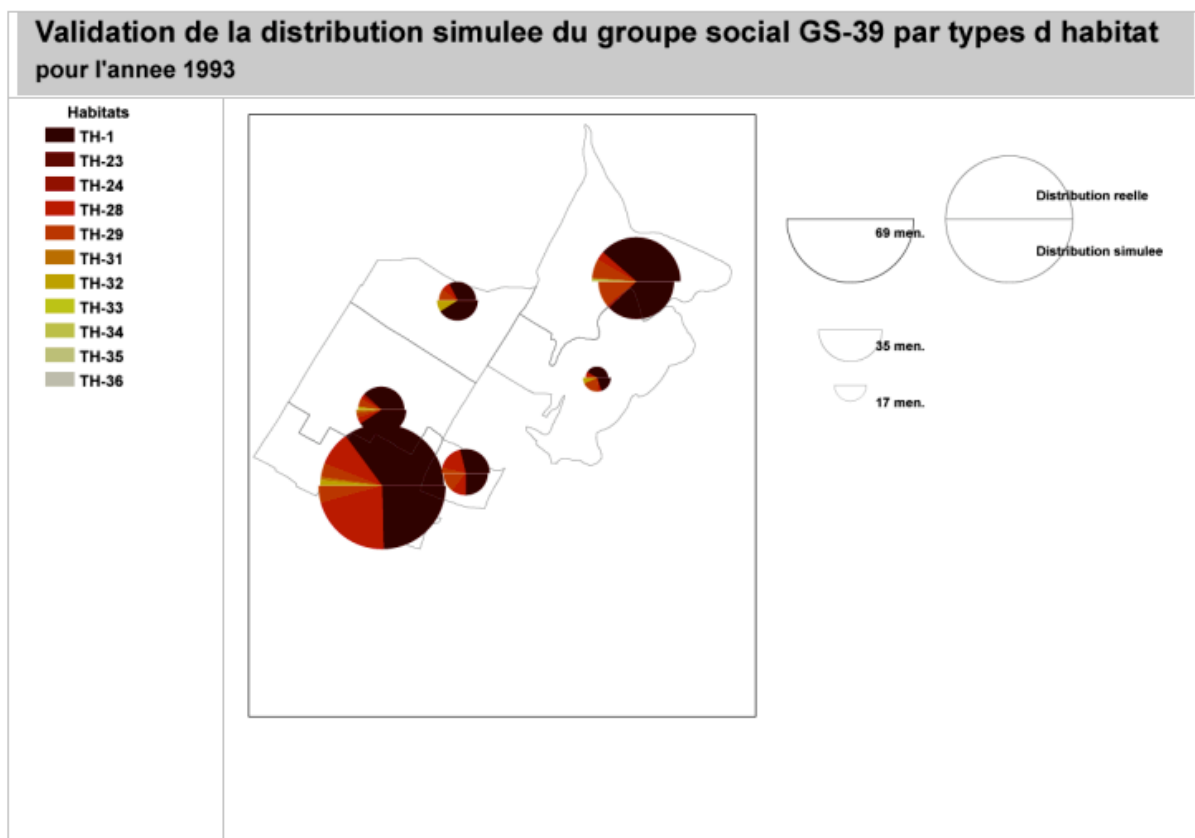


Figure 56. Exemple de carte de validation pour un groupe social. Validation du groupe social simulé *GS39* pour l'arrondissement « La Candelaria » année 1993.

### **9.3. Validation interne**

Pour faire une validation interne (validation stochastique des résultats) du mécanisme d'interaction nous avons effectué une batterie de simulations avec les mêmes données d'entrée (même distribution initiale et graphe d'interconnexion) et mêmes paramètres internes (par exemple pour le cas du modèle étendu, on a utilisé le même algorithme de classification automatique pour la formation des groupes) et on a étudié la variabilité du coefficient de corrélation de la distribution finale simulée par rapport à celui de la distribution finale réelle.

Nous avons aussi réalisé une analyse de sensibilité des résultats des simulations par rapport aux paramètres de mobilité (coûts de déménagement, facteur de prix pour les groupes sociaux non-localisés), au paramétrage de la formation des groupes (algorithme de classification utilisé et paramétrage de l'algorithme) et au mécanisme de construction des règles d'évolution. Les résultats de ces analyses sont présentés dans la partie III de ce document.

## **Partie III Application et Résultats**

Dans cette partie, nous présentons les résultats de plusieurs expériences réalisées avec les modèles présentés dans la partie II. Ces expériences ont été réalisées sur des données issues des recensements généraux de la population de Bogotá pour les années 1973 et 1993. Nous avons travaillé sur des secteurs urbains centraux de cette ville. Tout d'abord, nous décrivons le type de résultats obtenus à l'issue des simulations (Chapitre X) ainsi que leur post-traitement pour les afficher dans des formats facilement lisibles. Nous présentons ensuite les différentes expériences réalisées (Chapitre XI) et les résultats obtenus. Finalement, dans le chapitre XII nous discutons ces résultats.

## **Chapitre X : Production des Résultats**

Dans ce chapitre, nous présentons une description des fichiers de résultats produits par une simulation et un mécanisme de traitement de ces fichiers qui permet de produire des cartes et des tableaux récapitulatifs, pour ainsi donner une présentation plus facilement lisible des résultats obtenus et des relations entre les résultats.

### **10.1. Fichiers de résultats**

Chaque simulation produit plusieurs fichiers de résultats. Pour chaque année de simulation, on produit trois fichiers différents :

- le fichier *socialGroups.xml* qui contient la description et la distribution spatiale des groupes sociaux ;
- le fichier *habitats.xml* qui contient la description, la distribution spatiale et la liste d'occupation de chaque type d'habitat ;
- le fichier *Migrations.xml* qui contient une description des déménagements effectués dans l'année de simulation.

Additionnellement, pour chaque simulation, on produit le fichier *GroupsAliases.xml* (un seul par simulation) qui contient la mise en relation des groupes sociaux et des types d'habitat de toutes les années de simulation (cf. section 7.2.5).

#### **10.1.1 Description du fichier socialGroups.xml**

Le fichier *socialGroups.xml* contient la distribution spatiale et la description de tous les groupes sociaux pour une année de simulation. Ce fichier est décrit par la DTD<sup>12</sup> suivante :

---

<sup>12</sup> Une DTD (Document Type Definition) est un document permettant de décrire un modèle de document XML. Une DTD indique les noms des éléments qui peuvent apparaître et leur contenu : les sous-éléments et les attributs. En dehors des attributs, le contenu est spécifié en indiquant le nom, l'ordre et le nombre d'occurrences autorisées des sous-éléments (cf. <http://www.w3schools.com/dtd/default.asp>). Il existe un autre type plus

```

<!DOCTYPE SocialGroups [
  <!ELEMENT SocialGroups (SocialGroup*)>
  <!ELEMENT SocialGroup (ProfileDescription, Distribution)>
  <!ELEMENT Distribution (UrbanSectorDistribution*)>
  <!ELEMENT UrbanSectorDistribution (Occupation*)>

  <!ATTLIST SocialGroup id ID #REQUIRED>
  <!ATTLIST SocialGroup totalNumber CDATA #REQUIRED>
  <!ATTLIST SocialGroup unLocalisedNumber CDATA #REQUIRED>

  <!ATTLIST ProfileDescription averageProfile CDATA #REQUIRED>
  <!ATTLIST ProfileDescription size CDATA #REQUIRED>
  <!ATTLIST ProfileDescription histogramme CDATA #REQUIRED>
  <!ATTLIST ProfileDescription histogrammeLength CDATA #REQUIRED>

  <!ATTLIST UrbanSectorDistribution localisation IDREF #REQUIRED>
  <!ATTLIST UrbanSectorDistribution totalNumber CDATA #REQUIRED>

  <!ATTLIST Occupation id IDREF #REQUIRED>
  <!ATTLIST Occupation number CDATA #REQUIRED>
  <!ATTLIST Occupation preference CDATA #REQUIRED>
]>

```

*Encadré 14. DTD qui décrit la structure du fichier socialGroups.xml*

L'élément *SocialGroup* qui représente un groupe social contient les attributs suivants :

- l'attribut *id* qui contient l'identifiant unique du groupe<sup>13</sup>,
- l'attribut *totalNumber* qui contient le nombre d'effectifs qui composent le groupe,
- l'attribut *unLocalisedNumber* qui contient le nombre de ménages sans logement dans le groupe social représenté. Ce dernier élément est composé de deux sous-éléments : *Distribution* et *ProfileDescription*.

L'élément *ProfileDescription* contient des informations synthétiques qui décrivent le profil du groupe représenté. Ce sous-élément contient quatre attributs:

- l'attribut *histogramme* qui contient une liste d'entiers séparés par le caractère « : ». Chacune des valeurs dans l'histogramme représente le nombre d'effectifs dans une modalité donnée ;

---

restrictif de description des modèles de documents XML : le XSD (XML Schema Description ; cf. <http://www.w3.org/XML/Schema>).

<sup>13</sup> Le *id* d'un groupe social dans le fichier *socialGroup.xml* est une chaîne de caractères qui commence par « CLUST-» (*cluster*, classe en anglais), suivi par un entier unique par groupe.



- l'attribut *averageProfile* qui contient la description sous forme d'entier du profil moyen du groupe (cf. Annexe II, Représentation des profils). La représentation binaire de ce nombre entier représente le profil moyen du groupe sous forme disjonctive complète. Le vecteur binaire qui définit la représentation sous forme disjonctive complète est construit à partir des valeurs dans l'histogramme. Dans le profil moyen, une modalité est représentée par un 1 si le nombre d'effectifs dans la modalité est supérieur à la moitié du nombre total d'effectifs. Autrement, la modalité est représentée par un 0 dans le profil moyen.
- L'attribut *histogrammeLength* contient le nombre de modalités considérées dans l'histogramme.
- L'attribut *size* contient le nombre d'effectifs dans le groupe.

L'élément *Distribution* représente la distribution spatiale du groupe social. Cet élément est composé de plusieurs sous-éléments *UrbanSectorDistribution*. Chacun de ces sous-éléments représente la liste d'occupation du groupe social dans un secteur urbain donné. Un élément *UrbanSectorDistribution* contient le nombre de logements occupés (attribut *totalNumber*) par un groupe social dans un secteur urbain donné (attribut *localisation*). À l'intérieur de cet élément se trouve une liste d'éléments *Occupation*. Chacun de ces éléments contient le nombre de logements (attribut *number*) d'un type d'habitat donné (attribut *id*) occupés par le groupe social. L'attribut *preference* contient le niveau de préférence du groupe social pour le type d'habitat donné dans le secteur urbain donné.

Dans l'encadré ci-dessous nous présentons un exemple des éléments les plus importants qui composent un fichier *socialGroups.xml*.

```
<SocialGroups>
    ...
    <SocialGroup id="CLUST-22947" totalNumber="1542" unLocalisedNumber="23" >
        <ProfileDescription averageProfile="4829691847168" size="1542"
        histogramme="1516:26:34:10:1498:1524:18:72:1470:11:219:837:402:73:185:232:305:235:181:52:352:
        364:1041:105:32:44:1498:398:180:964:21:133:32:1279:77:182:135:187:186:258:285:212:97"
        histogrammeLength="43"/>
        <Distribution>
            <UrbanSectorDistribution localisation="S_URBAN=3101" totalNumber="73" >
                <Occupation id="CLUST-23567" number="20" preference="0.273973" />
                <Occupation id="CLUST-23568" number="3" preference="0.0410959" />
                ...
                <Occupation id="CLUST-23572" number="10" preference="0.136986" />
                <Occupation id="CLUST-23571" number="24" preference="0.328767" />
            </UrbanSectorDistribution>
        </Distribution>
    </SocialGroup>
</SocialGroups>
```

```

        </UrbanSectorDistribution>
        ...
    </Distribution>
</SocialGroup>
    ...
</SocialGroups>

```

Encadré 15. Extrait d'un fichier socialGroups.xml produit par une simulation

### 10.1.2 Description du fichier habitats.xml

Le fichier *habitats.xml* représente les matrices d'occupation de chaque type d'habitat. Il contient aussi la description de chaque type d'habitat. Le fichier *habitats.xml* est décrit par la DTD suivante :

```

<!DOCTYPE Habitats [
  <!ELEMENT Habitats (Habitat*)>
  <!ELEMENT Habitat (ProfileDescription, Distribution)>
  <!ELEMENT Distribution (UrbanSectorDistribution*)>
  <!ELEMENT UrbanSectorDistribution (Occupation*)>

  <!ATTLIST Habitat id ID #REQUIRED>
  <!ATTLIST Habitat totalNumber CDATA #REQUIRED>
  <!ATTLIST Habitat freeNumber CDATA #REQUIRED>

  <!ATTLIST ProfileDescription averageProfile CDATA #REQUIRED>
  <!ATTLIST ProfileDescription size CDATA #REQUIRED>
  <!ATTLIST ProfileDescription histogrammeLength CDATA #REQUIRED>
  <!ATTLIST ProfileDescription histogramme CDATA #REQUIRED>

  <!ATTLIST UrbanSectorDistribution localisation IDREF #REQUIRED>
  <!ATTLIST UrbanSectorDistribution totalNumber CDATA #REQUIRED>
  <!ATTLIST UrbanSectorDistribution freeNumber CDATA #REQUIRED>

  <!ATTLIST Occupation id IDREF #REQUIRED>
  <!ATTLIST Occupation number CDATA #REQUIRED>
]>

```

Encadré 16. DTD qui décrit la structure du fichier habitats.xml

L'élément *Habitat* qui représente un type d'habitat contient les attributs suivants :

- l'attribut *id* qui contient l'identifiant unique du type d'habitat,

- l'attribut *totalNumber* qui contient le nombre d'effectifs qui composent le type d'habitat,
- l'attribut *freeNumber* qui contient le nombre total de logements libres du type d'habitat décrit.

L'élément *Habitat* est composé de deux sous éléments : *ProfileDescription* et *Distribution*. L'élément *ProfileDescription* est similaire à l'élément du même nom présenté dans la DTD qui définit la structure du fichier *socialGroups.xml*.

L'élément *Distribution* contient la distribution géographique ainsi que les listes d'occupation du type d'habitat représenté. Cet élément contient une liste d'éléments *UrbanSectorDistribution*. Chaque élément *UrbanSectorDistribution* contient le nombre de logements du type d'habitat décrit (attribut *totalNumber*) et le nombre de logements libres (attribut *freeNumber*) dans un secteur urbain donné (attribut *localisation*). À l'intérieur de chaque élément *UrbanSectorDistribution*, on trouve une liste d'éléments *Occupation*. Chacun de ces éléments contient le nombre de logements (attribut *number*) du type d'habitat représenté, occupé par un certain groupe social (attribut *id*).

Dans l'encadré ci-dessous, nous présentons à manière d'exemple, un extrait d'un fichier *habitats.xml* produit par une simulation.

```

<Habitats>
...
  <Habitat id="CLUST-35362" totalNumber="4339" freeNumber="49">
    <ProfileDescription averageProfile="289573206" size="4339"
      histogramme="2392:1646:36:265:3697:642:4270:69:127:327:757:3075:53:4271:16:0:52:4333
      :6:6:4333:51:4288:69:4270:72:4267:4267:72" histogrammeLength="29"/>
    <Distribution>
      <UrbanSectorDistribution localisation="S_URBAN=3101" totalNumber="108" freeNumber="3">
        <Occupation id="CLUST-34728" number="5" />
        <Occupation id="CLUST-34732" number="8"/>
        ...
        <Occupation id="CLUST-34738" number="7"/>
        <Occupation id="CLUST-34726" number="13"/>
      </UrbanSectorDistribution>
      ...
    </Distribution>
  </Habitat>
  ...
</Habitats>

```

Encadré 17. Extrait d'un fichier *habitats.xml* produit par une simulation

### 10.1.3 Description du fichier Migrations.xml

Le fichier Migrations.xml contient une liste des déménagements effectués sur une année de simulation. Le fichier Migrations.xml est décrit par la DTD suivante :

```
<!DOCTYPE Migrations [  
  <!ELEMENT Migrations (Migration*)>  
  
  <!ATTLIST Migration socialGroup IDREF #REQUIRED>  
  <!ATTLIST Migration originArea IDREF "OUTSIDE">  
  <!ATTLIST Migration originHabitat IDREF "OUTSIDE">  
  <!ATTLIST Migration destinationArea IDREF #REQUIRED>  
  <!ATTLIST Migration destinationHabitat IDREF #REQUIRED>  
  <!ATTLIST Migration size CDATA #REQUIRED>  
>
```

Encadré 18. DTD qui décrit la structure du fichier Migrations.xml

Dans cette définition, chaque élément *Migration* représente le déménagement d'un ensemble de ménages d'un même groupe social. Il contient six attributs:

- *socialGroup* qui contient l'id du groupe social qui déménage ;
- *originArea* qui contient le nom du secteur urbain qui est le point de départ du déménagement ;
- *originHabitat* qui contient l'id du type d'habitat occupé avant le déménagement ;
- *destinationArea* qui contient le nom du secteur urbain qui est le point d'arrivée du déménagement ;
- *destinationHabitat* contient l'id du type d'habitat occupé après le déménagement ;
- *size* qui contient le nombre de ménages qui ont déménagé.

Le mot clé « OUTSIDE » est utilisé dans les attributs *originArea* et *originHabitat* pour représenter des déménagements des groupes sociaux non-localisés (cf. section 6.2.2.2.). Ci-dessous nous présentons un extrait d'un fichier *Migrations.xml* produit par une simulation.

```
<Migrations>  
  <Migration socialGroup="CLUST-70424" originArea="OUTSIDE" originHabitat="OUTSIDE"  
    destinationArea="S_URBAN=3210" destinationHabitat="CLUST-71104" size="1"/>  
  <Migration socialGroup="CLUST-70421" originArea="OUTSIDE" originHabitat="OUTSIDE"  
    destinationArea="S_URBAN=8109" destinationHabitat="CLUST-71103" size="3"/>  
  <Migration socialGroup="CLUST-70406" originArea="S_URBAN=3102" originHabitat="CLUST-  
    71107" destinationArea="S_URBAN=8105" destinationHabitat="CLUST-71103" size="1"/>  
  <Migration socialGroup="CLUST-70406" originArea="S_URBAN=3212" originHabitat="CLUST-  
    71102" destinationArea="S_URBAN=8105" destinationHabitat="CLUST-71103" size="1"/>
```

```
...  
</Migrations>
```

Encadré 19. Extrait d'un fichier Migrations.xml produit par une simulation

#### 10.1.4 Fichier GroupsAliases.xml

Le fichier *GroupsAliases.xml* contient la mise en relation des ensembles de groupes sociaux et des ensembles de types d'habitat tout au long de la simulation (cf. section 7.2.5.). Les informations contenues dans ce fichier permettent de retracer l'historique de chaque groupe social et de chaque type d'habitat. La structure du fichier *GroupsAliases.xml* est décrite par la DTD suivante :

```
<!DOCTYPE Aliases [  
<!ELEMENT Aliases (SocialGroups, Habitats)>  
<!ELEMENT SocialGroups(Group*)>  
<!ELEMENT Habitats(Group*)>  
<!ELEMENT Group(Iteration*)>  
  
<!ATTLIST Group id IDREF #REQUIRED >  
<!ATTLIST Iteration iterationName CDATA #REQUIRED >  
<!ATTLIST Iteration groupName CDATA #REQUIRED >  
>
```

Encadré 20. DTD qui décrit la structure du fichier GroupsAliases.xml

Dans cette DTD, l'élément *SocialGroups* contient les mises en relation pour les groupes sociaux. L'élément *Habitats* contient les mises en relation pour les types d'habitat. Ces deux éléments contiennent des sous-éléments de type *Group*. Chaque élément *Group* contient la mise en relation d'un groupe social ou d'un type d'habitat. Cet élément est défini par l'attribut *id*. L'attribut *id* contient un nom unique généré de manière automatique suivant les formats suivants :

- Dans le cas des groupes sociaux, *id* contient une chaîne de caractères qui commence par le prefix « GS-» suivi d'un numéro entier qui est unique pour chaque groupe social représenté.
- Dans le cas des types d'habitat, *id* contient une chaîne de caractères qui commence par le prefix « TH-» suivi d'un numéro entier qui est unique pour chaque type d'habitat représenté.

Chaque élément *Group* est composé de plusieurs sous-éléments *Iteration*. Chacun des éléments *Iteration* contient le nom (attribut *groupName*) du groupe social ou du type d'habitat

représenté à une année de simulation donnée (attribut *iterationName*). L'attribut *groupName* contient l'id du groupe social ou du type d'habitat dans les fichiers *socialGroups.xml* ou *habitats.xml* de l'itération (année de simulation) correspondante. L'attribut *iterationName* peut contenir soit un entier qui varie entre 0 et le nombre d'années de simulation – 1, soit le mot « Init », soit le mot « Validation ». Le mot « Init » veut dire qu'il s'agit de l'année d'initialisation (distribution de la population initiale réelle). Le mot « Validation » veut dire qu'il s'agit de l'année de validation (distribution de la population finale réelle). Lorsque l'attribut *iterationName* contient un entier, il représente une années de simulation. Ci-dessous nous présentons un extrait d'un fichier *GroupsAliases.xml* produit par une simulation. On remarque dans cet exemple, que certains groupes sociaux ou types d'habitat disparaissent avant la dernière année de simulation (par exemple le groupe social GS-0) et d'autres apparaissent en milieu de la simulation (par exemple le type d'habitat TH-).

```

<Aliases>
  <SocialGroups>
    <Group id="GS-0">
      <Iteration groupName="CLUST-11155" name="Init"/>
      <Iteration groupName="CLUST-22937" name="0"/>
      <Iteration groupName="CLUST-34743" name="1"/>
      ...
      <Iteration groupName="CLUST-117209" name="8"/>
      <Iteration groupName="CLUST-128875" name="9"/>
    </Group>
    <Group id="GS-1">
      <Iteration groupName="CLUST-11153" name="Init"/>
      <Iteration groupName="CLUST-22949" name="0"/>
      <Iteration groupName="CLUST-34734" name="1"/>
      ...
      <Iteration groupName="CLUST-248203" name="19"/>
      <Iteration groupName="CLUST-262260" name="Validation"/>
    </Group>
    ...
  </SocialGroups>
  <Habitats>
    ...
    <Group id="TH-2" name="CLUST-11777">
      <Iteration groupName="CLUST-11777" name="Init"/>
      <Iteration groupName="CLUST-23572" name="0"/>
      ...
  </Habitats>
</Aliases>

```

```

        <Iteration groupName="CLUST-235155" name="18"/>
        <Iteration groupName="CLUST-249208" name="19"/>
        <Iteration groupName="CLUST-263265" name="Validation"/>
    </Group>
    ...
    <Group id="TH-10" name="CLUST-94323">
        <Iteration groupName="CLUST-94323" name="6"/>
        <Iteration groupName="CLUST-106108" name="7"/>
        <Iteration groupName="CLUST-117852" name="8"/>
        <Iteration groupName="CLUST-129538" name="9"/>
        ...
        <Iteration groupName="CLUST-235153" name="18"/>
        <Iteration groupName="CLUST-249212" name="19"/>
        <Iteration groupName="CLUST-263269" name="Validation"/>
    </Group>
    ...
</Habitats>
...
</Aliases>

```

Encadré 21. Extrait d'un fichier GroupsAliases.xml produit par une simulation

## 10.2. Traitement des résultats

La représentation des résultats sous les fichiers XML décrits dans la section précédente, facilite leur traitement a posteriori. Le processus principal de traitement des résultats est présenté dans la figure ci-dessous. Par ce processus, nous produisons des cartes et des tableaux récapitulatifs, à partir de la transformation des fichiers de résultats. Ces transformations sont faites à l'aide de différentes feuilles de style *XSL*<sup>14</sup> (cf. Annexe VI). Ces transformations sont faites par un moteur de transformation *XSLT*<sup>15</sup>. Le moteur de transformation *XSLT* utilisé est *Xalan-Java* version 2.7.0 développé par la fondation Apache

<sup>14</sup> *XSL* ( *eXtended Style Language*, cf. <http://www.w3.org/Style/XSL/>) est un langage de description des feuilles de style associé à XML. Une feuille de style *XSL* est un fichier qui décrit comment doivent être présentés les documents XML basés sur une même DTD ou schema XML.

<sup>15</sup> *XSLT* ( *eXtended Stylesheet Language Transformations*, cf. ) est le langage de transformation associé aux feuilles de style *XSL*. Il permet de transformer un document XML vers un autre, mais aussi vers un autre type de document tels qu'un document de texte ou pdf.

(cf. <http://xml.apache.org/xalan-j/>) avec l'extension *saxon 8.0* pour Java développée par Michale Kay (cf. <http://www.saxonica.com/>). L'exécution de la transformation des résultats se fait par l'intermédiaire de différents *scripts* (cf. Annexe VII) qui reçoivent en paramètre la feuille de style XSL et les répertoires d'entrée et sortie, ainsi que d'autres paramètres propres à chaque transformation (par exemple la carte de la ville).

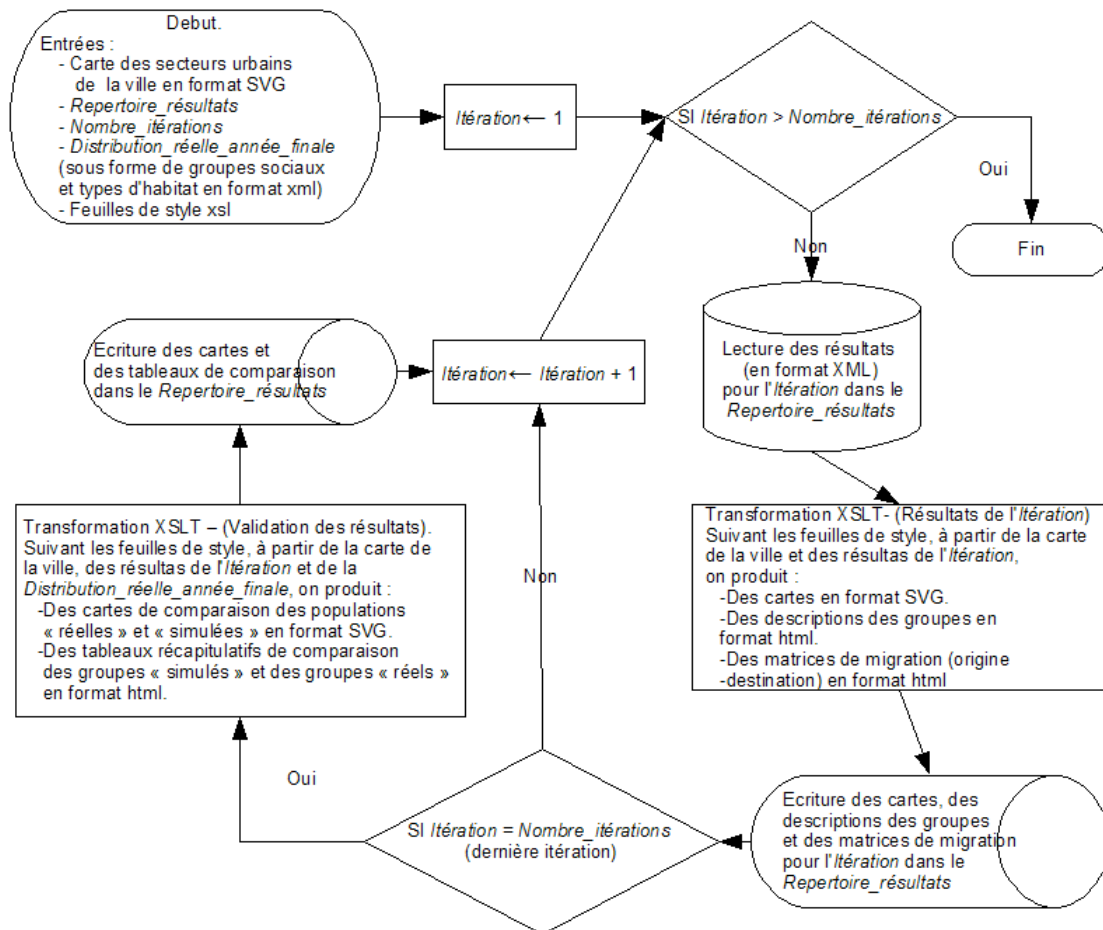


Figure 57. Transformation des résultats d'une simulation.

La transformation des résultats peut être changée facilement par la modification ou la création des nouvelles feuilles de style XSL. Nous avons programmé les feuilles de style présentées dans l'Annexe VI. Dans la suite, nous présentons les principales transformations actuellement programmées.

### 10.2.1. Production de la cartographie des résultats

La feuille de style *SVGMap.xsl* (cf. Annexe A6.1) permet la production des cartes de distribution des types d'habitat, des groupes sociaux ainsi que des cartes de comparaison entre les matrices d'occupation « *simulées* » et les matrices d'occupation « *réelles* ». Cette



comparaison est faite entre les groupes détectés sur la population et le parc de logements de la dernière itération de la simulation et les groupes détectés sur la population et le parc de logements réels. Des exemples de ces cartes sont présentés dans la Annexe A.5.3.

Les cartes sont produites dans un format *SVG*<sup>16</sup>. Pour la production de ces cartes, il est nécessaire d'avoir en entrée une carte des secteurs urbains dans un format *SVG*. Ce format est affichable dans n'importe quel navigateur WEB sous la condition de l'installation d'un *plugin* adapté (par exemple, le *Adobe-SVG-viewer*)<sup>17</sup>. D'autres programmes *stand-alone* permettent aussi la visualisation et la transformation de fichiers en format *SVG*, tel est le cas de différents composants graphiques développés en java sur la librairie batik<sup>18</sup>.

La structure du fichier contenant la carte d'entrée est décrite par la DTD suivante :

```
<!ELEMENT svg (g*)>
<!ELEMENT g (path)>

<!ATTLIST g urbanSector ID #REQUIRED>
<!ATTLIST g rmax CDATA #REQUIRED>
<!ATTLIST g rmin CDATA #REQUIRED>
<!ATTLIST g xcenter CDATA #REQUIRED>
<!ATTLIST g ycenter CDATA #REQUIRED>
```

Encadré 22. DTD qui décrit la structure du fichier contenant la carte des secteurs urbains de la ville

Chaque élément *g* contient dans le sous-élément *path* les contours d'un secteur urbain donné.

Les attributs exigés dans l'élément *g* sont :

- *urbanSector* qui est l'identifiant du secteur urbain représenté,
- *xcenter* et *ycenter* qui contiennent les coordonnées du centre de gravité du secteur urbain représenté,
- *rmax* et *rmin* contiennent les distances maximale et minimale entre le centre de gravité et le contour du secteur urbain donné.

#### 10.2.1.1. Cartes de distribution spatiale

À partir de la carte de secteurs urbains et les fichiers *socialGroup.xml* et *habitats.xml* d'une itération donnée, nous produisons une carte de la distribution spatiale par groupe social et une carte de la distribution spatiale par type d'habitat. Dans chacune de ces cartes, les éléments

<sup>16</sup> SVG ( *Scalable Vector Graphics*, cf. ) est un format de fichier basé sur XML qui permet de décrire des ensembles de graphiques vectoriels.

<sup>17</sup> cf. <http://www.adobe.com/svg/viewer/install/main.html>

<sup>18</sup> cf. <http://xmlgraphics.apache.org/batik/index.html>

Occupation des fichiers *socialGroup.xml* et *habitats.xml* sont transformés en graphiques sous forme de camemberts par secteur urbain (cf. section 7.2.5.).

### 10.2.1.2. Cartes de validation

À partir du fichier *socialGroups.xml* de la dernière itération et le fichier *socialGroups.xml* de l'année de validation (distribution finale réelle), nous produisons des cartes de validation de la distribution pour chaque groupe social. Dans chaque carte, nous présentons sous forme de demi-camemberts les distributions simulées et réelles. (cf. section 9.2.).

### 10.2.1.3. Production de tableaux récapitulatifs de migration

Les migrations entre secteurs urbains sont représentées sous forme de matrices d'origine-destination. Ces matrices sont produites par la transformation des fichiers *Migrations.xml* de chaque itération à l'aide de la feuille de style *Migrations.xsl* (cf. Annexe A6.3). De même, des tableaux récapitulatifs de la composition des groupes générés sont produits à l'aide de la même feuille de style. Ces tableaux et matrices sont produits en format *html*. (Voir exemple ci-dessous)

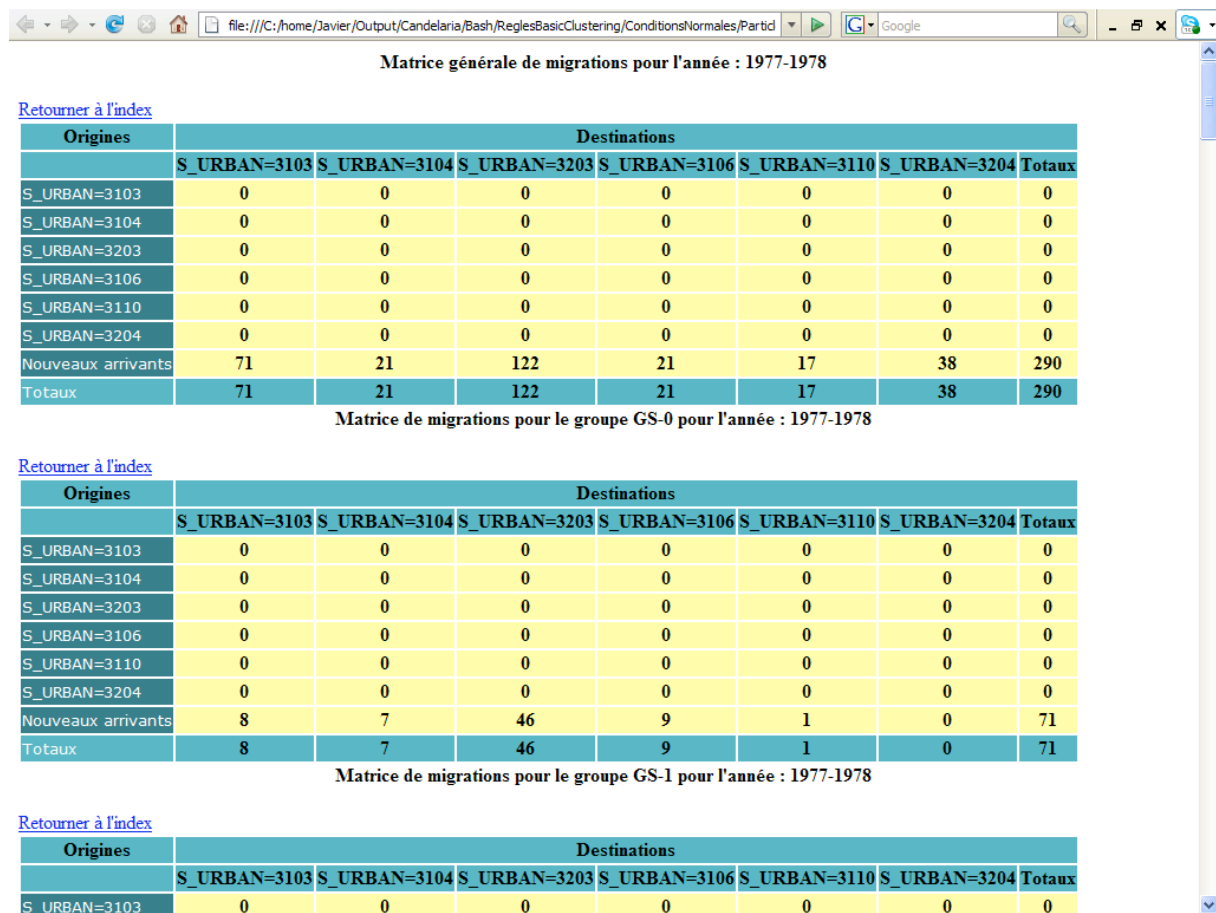


Figure 58. Exemple de tableaux de migration, pour l'année de simulation 1977-1978

## 10.2.2. Description automatique des groupes sociaux et des types d'habitat

À partir de l'attribut *histogramme* de l'élément *ProfilDescription* qui contient le profil moyen des groupes sociaux et des types d'habitat sous forme d'histogramme, nous produisons une description automatique basée sur le calcul des valeurs test (cf. 3.1.3.2.). Nous produisons ainsi des graphiques contenant les modalités de chaque caractéristique dont les *valeurs-test*<sup>19</sup> sont maximales (voir exemple ci-dessous).

Caractérisation automatique du groupe GS-6 pour l'année 1985 (population simulée)  
Selon les valeurs de test

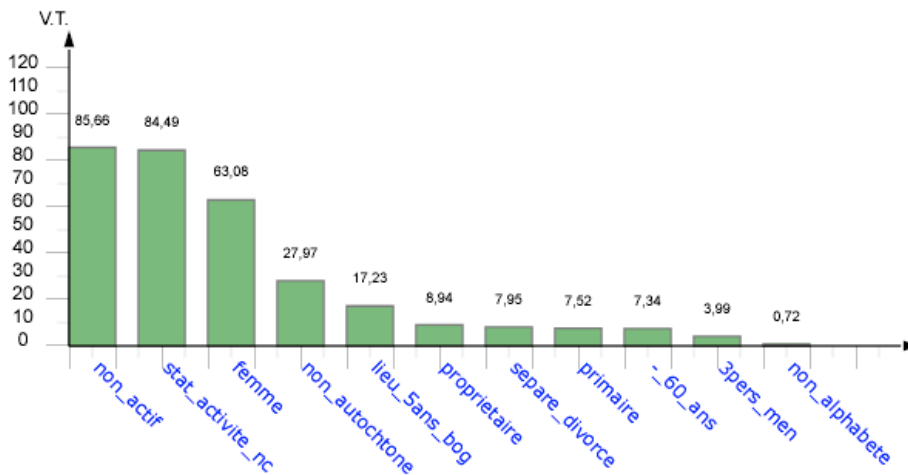


Figure 59. Exemple de graphique de description automatique d'un groupe social ou d'un type d'habitat.

À partir de l'histogramme, nous produisons aussi des tableaux récapitulatifs de la description de chaque groupe social et de chaque type d'habitat. Ces tableaux récapitulatifs contiennent les valeurs tests, ainsi que les différentes statistiques utilisées dans le calcul de ces valeurs test (voir exemple ci-dessous).

<sup>19</sup> Une valeur-test est un critère qui permet d'apprécier rapidement si une modalité a une position « significative » dans une classe (cf Lebart et al, 2006). Elle est significative au seuil usuel 5% si elle dépasse la valeur 1,96 : l'hypothèse "nulle" est rejetée et la moyenne ou la proportion d'une modalité sur la population globale et celle dans la classe diffèrent significativement

Simulation Results

file:///Users/javier/Output/Candelaria/Bash/DecisionMesoscopique/Con Google

Le site Mozilla Mozilla en français

(CLUST-121043) 1095 13,24%

Analyse de la distribution des modalités pour le groupe : GS-0(CLUST-121040)  
 Pour l'année 1989(population simulée)  
 Nombre d'effectifs : 1385 ( 19,31% )sur une population de individus

[Retour à l'index des groupes](#)

Variable	Modalités	Effectifs dans la modalité	% modalité dans le group par rapport au group. Les pourcentages en rouge sont plus grands que le pourcentage de la modalité dans la population	% modalité dans le group par rapport à la population	% de la modalité dans la population	Valeur test
lieu_de_residence_5_ans_avant_lenquete	lieu_5ans_bog	1320	95,31%	21,52%	85,53%	11,52
	lieu_5_ans_ailleurs	65	4,69%	6,26%	14,47%	-11,52
W_semaine_precedente_en_classes	W_semaine_preced	1318	95,16%	25,17%	73,02%	20,66
	recherche_emploi	67	4,84%	19,65%	4,76%	0,16
	non_actif	0	0,00%	0,00%	22,23%	-22,14
Sexo	homme	1297	93,65%	27,63%	65,47%	24,53
	femme	88	6,35%	3,55%	34,53%	-24,53
Sabe_leer_en_classes	alphabete	1350	97,47%	19,51%	96,47%	2,25
	non_alphabete	35	2,53%	13,83%	3,53%	-2,25
Posicion_categoria_en_el_trabajo_en_classes	independant	697	50,32%	44,23%	21,98%	28,35
	patron	160	11,55%	26,53%	8,41%	4,68
	emploi_domestique	64	4,62%	26,56%	3,36%	2,90
	ouvrier_employe	438	31,62%	14,60%	41,84%	-8,57
	stat_activite_nc	26	1,88%	1,48%	24,42%	-21,73
Numero_de_personas_en_el_hogar_en_classes	+7pers_men	180	13,00%	27,03%	9,29%	5,28
	1pers_men	312	22,53%	21,89%	19,87%	2,76
	6pers_men	103	7,44%	21,64%	6,64%	1,33
	5pers_men	165	11,91%	20,60%	11,17%	0,97
	4pers_men	216	15,60%	18,45%	16,33%	-0,82
	3pers_men	200	14,44%	15,66%	17,81%	-3,65
	2pers_men	209	15,09%	15,42%	18,90%	-4,03
Nivel_Educativo_en_classes	primaire	627	45,27%	24,03%	36,38%	7,65
	secondaire	514	37,11%	18,64%	38,46%	-1,15
	sans_niv_educ	55	3,97%	16,72%	4,59%	-1,22
	superieur	189	13,65%	12,81%	20,57%	-7,09

Terminé

Figure 60. Exemple d'un tableau récapitulatif de caractérisation automatique d'un groupe social.

La production des tableaux et graphiques de description automatique des groupes sociaux et des types d'habitat est définie par la feuille de style *Statistics.xml* (cf. Annexe A6.2). Pour pouvoir produire ces résultats, la feuille de style a besoin de connaître l'ordre des modalités et des caractéristiques des ménages et des logements dans un histogramme. Cet ordre est défini dans deux fichiers : *HouseholdsMapping.xml* et *HousingsMapping.xml*. Le premier fichier contient la liste des caractéristiques et des modalités qui décrivent les profils des ménages, le deuxième fichier contient la liste des caractéristiques et des modalités qui décrivent les profils des logements. Le format de ces deux fichiers est défini par la DTD ci-dessous. Dans cette DTD, les caractéristiques sont représentées par les éléments *Variable*. L'attribut *name* de ces

éléments contient le nom de la caractéristique représentée. Chaque élément *Variable* est composé de plusieurs éléments *Modality* qui représentent les différentes modalités de chaque caractéristique. L'attribut *name* de l'élément *Modality* contient le nom de la modalité. L'attribut *order* dans les éléments *Variable* et *Modality* définit la localisation relative de chaque variable et de chaque modalité dans un profil sous forme disjonctive complète. L'attribut *order* des caractéristiques (*Variable*) prend des valeurs entre 1 et le nombre de caractéristiques. L'attribut *order* des modalités prend des valeurs entre 1 et le nombre de modalités pour chaque caractéristique. Les fichiers *HouseholdsMapping.xml* et *HousingsMapping.xml* sont produits par un processus automatique de pré-traitement des données présenté dans le chapitre suivant.

```

<!DOCTYPE Variables [
<ELEMENT Variables (Variable *)>
<ELEMENT Variable (Modality*)>

<!ATTLIST Variables entity CDATA #REQUIRED >
<!ATTLIST Variable name CDATA #REQUIRED >
<!ATTLIST Modality name CDATA #REQUIRED >
]>

```

Encadré 23. DTD qui décrit la structure des fichiers *HouseholdsMapping.xml* et *HousingsMapping.xml*

Dans les encadrés ci-dessous, nous montrons des extraits des fichiers *HouseholdsMapping.xml* et *HousingsMapping.xml*.

### 10.2.3. Production des diagrammes historiques de groupes et de types d'habitat

L'évolution des groupes sociaux et des types d'habitat est représentée sous forme synthétique à l'aide des « histogrammes d'évolution » (voir un exemple dans la figure ci-dessous). La production de ces histogrammes est définie dans la feuille de calcul *Statistics.xml* (cf. Annexe A6.2).

## Evolution historique de la taille des types d'habitats

	TH-0	TH-1	TH-2	TH-3	TH-4	TH-5	TH-6	TH-7	TH-8	TH-9	TH-10	TH-11
Init	303	1549	1259	1660	5502	6637	84	1435	3716			
1974		1564	1420	1688	3875	7777	87	1214	4878			
1975		1733	811	2353	3450	8402	118	1524	4339			
1976		2880	486	2673	5587	6580	121	394	2623	1464		
1977		600	776	1746	3146	6859	51		5150	4582		
1978		726	3054	4457	98	3038	84			11602		
1979		3088	1680	1409	59	6851	92			10036		
1980		1825	2117	1105	59	5128	96			8381	4705	
1981		39	1788	4800	69	5123	3751			7851	192	
1982		43	2363	5627	61	5198	3226			7067	208	
1983			2297	8692	41	6689	70			5912	196	
1984			4410	8470	55	3977	81			6936	194	
1985			3159	9189	43	4246	63			7466	193	
1986			3224	9212	43	4264	73			7596	220	
1987			3260	9304	46	4290	90			7677	228	
1988			3142	11358	37	4672	98			5618	227	
1989			3171	11486	40	4695	110			5692	221	
1990			1825	4738	49	8778	113			8518	186	
1991			2478	6384	65	7798	690			8301	164	
1992			2514	6430	69	7821	675			8415	169	
1993			2853	6645	89	7270	650			8344	347	
Validation			2853	6645	89	7270	650			8344	347	1496

Figure 61. Exemple d'histogramme d'évolution pour les types d'habitat.

Les histogrammes d'évolution sont de deux types : des histogrammes d'évolution générale et des histogrammes d'évolution par groupe. Dans les histogrammes d'évolution générale, nous représentons deux informations : l'évolution de la taille des groupes sociaux et des types d'habitat, et leur cycle de vie (année d'apparition, durée et année de disparition). Dans les histogrammes d'évolution par groupe, nous présentons l'évolution de la taille des groupes sociaux et types d'habitat par secteur urbain. Dans ces histogrammes, nous présentons aussi pour les groupes sociaux l'évolution du nombre de ménages sans logement et pour les types d'habitat, l'évolution du nombre de logements libres par secteur urbain.

### 10.3. Résultats de validation

Afin de valider les résultats des simulations, nous calculons les coefficients de corrélation entre les distributions des populations simulées et réelles (cf. Chapitre IX). Dans les résultats des simulations, nous présentons aussi les valeurs des coefficients de Davis-Bouldin pour chaque partition de ménages et de logements obtenues. Ces coefficients nous permettent d'avoir une idée de la qualité des classifications réalisées au cours des simulations pour construire les groupes sociaux et les types d'habitat.

## Chapitre XI : Application à Bogota

### 11.1. Objectifs des Expériences

Nous avons plusieurs objectifs dans la réalisation des simulations décrites dans ce chapitre. En premier lieu, il s'agit de déterminer quel modèle (modèle synthétique ou modèle étendu) est le plus adéquat pour modéliser l'évolution de la distribution spatiale des groupes sociaux dans certains quartiers de la ville de Bogotá (cf. section 11.2.1.1.). Le deuxième objectif est de définir l'impact de la participation des ménages à la prise de décision dans le modèle étendu. Enfin, il nous faut définir le paramétrage adéquat pour chacun des ces modèles. Nous considérons trois types de paramètres :

- **Paramètres de mobilité** : Ces paramètres font partie de la définition du mécanisme de mobilité au niveau mésoscopique. Nous considérons deux paramètres de mobilité : *la stratégie de mobilité* et *l'évolution de l'urgence pour résoudre la demande de logements*. La stratégie de mobilité se base sur la définition des coûts de déménagement (voir cas de figure présentés dans 11.2.3.1.1.). L'évolution de l'urgence pour résoudre la demande de logements se base sur la définition du facteur (*facteurPrix*) utilisé pour augmenter le prix proposé lors d'une enchère par un groupe social-non-localisé (cf. section 6.2.2.2.2.). Quant au choix de la stratégie de mobilité, nous voulons savoir si les ménages préfèrent déménager à l'intérieur de leur arrondissement courant (cf. cas 1 section 11.2.3.1.1), s'ils préfèrent déménager vers des arrondissements différents de leur arrondissement courant (cf. cas 2 section 11.2.3.3.1); ou bien s'il n'existe pas de stratégie particulière liée à la préférence pour un tel ou tel arrondissement (cf. cas 3 section 11.2.3.1.1). Dans la définition de l'urgence pour résoudre la demande, nous voulons établir quelques règles dans la définition du *facteurPrix*. Nous définissons ces règles à partir de l'analyse de la sensibilité des résultats des simulations à la variation des valeurs du *facteurPrix* (cf. 11.2.3.1.2).
- **Paramètres de formation de groupes** : Dans le cas du modèle étendu, le mécanisme de formation de groupes intervient au niveau de la constitution des groupes

sociaux et des types d'habitat initiaux et du mécanisme d'évolution des groupes sociaux et des types d'habitat (cf. section 7.2). Dans le cas du modèle synthétique, la formation des groupes intervient dans la constitution des groupes sociaux et des types d'habitat pour les années de simulation initiale et finale. Ce mécanisme repose sur un algorithme de classification. Le paramétrage du mécanisme de formation de groupes consiste alors à choisir l'algorithme de classification le plus adéquat et les conditions d'utilisation les plus adéquates. Ces conditions sont définies par le paramétrage propre à l'algorithme (cf. chapitre VIII). Une des conditions essentielles est le nombre de classes que l'algorithme doit trouver. La définition de ce paramètre est lié directement au degré d'abstraction souhaité. En effet, plus le nombre de classes est petit, plus le niveau d'abstraction et la distance avec le niveau microscopique est importante. Ainsi, nous nous intéressons à la définition du degré d'abstraction adéquat en fonction du nombre de classes trouvées par l'algorithme de classification (cf. section 11.2.3.2.).

- **Paramètres de construction des règles d'évolution :** La construction automatique des règles d'évolution présentées dans la section 7.2.3. dépend principalement de l'algorithme de classification utilisé pour construire les matrices de distribution de population initiale et finale (cf. section 7.2.3.5). La question est similaire à celle posée pour les paramètres de formation des groupes : quel est le niveau d'abstraction adéquat pour la construction de ces règles ? Nous donnons une réponse à partir des résultats obtenus à l'issue de plusieurs simulations décrites dans la section 11.2.4.

Dans la suite de ce chapitre, nous présentons différents types de simulation (cf section 11.2). Les résultats de ces simulations (cf. section 11.3) nous permettent d'atteindre les objectifs en donnant une réponse aux questions posées dans cette section.

## 11.2. Description des Expériences

### 11.2.1 Description des données

#### 11.2.1.1. Populations considérées

Nous avons réalisé plusieurs expériences sur les données de deux arrondissements du centre traditionnel de Bogota : « La Candelaria » et « Santa Fé » (voir carte ci-dessous). Les données



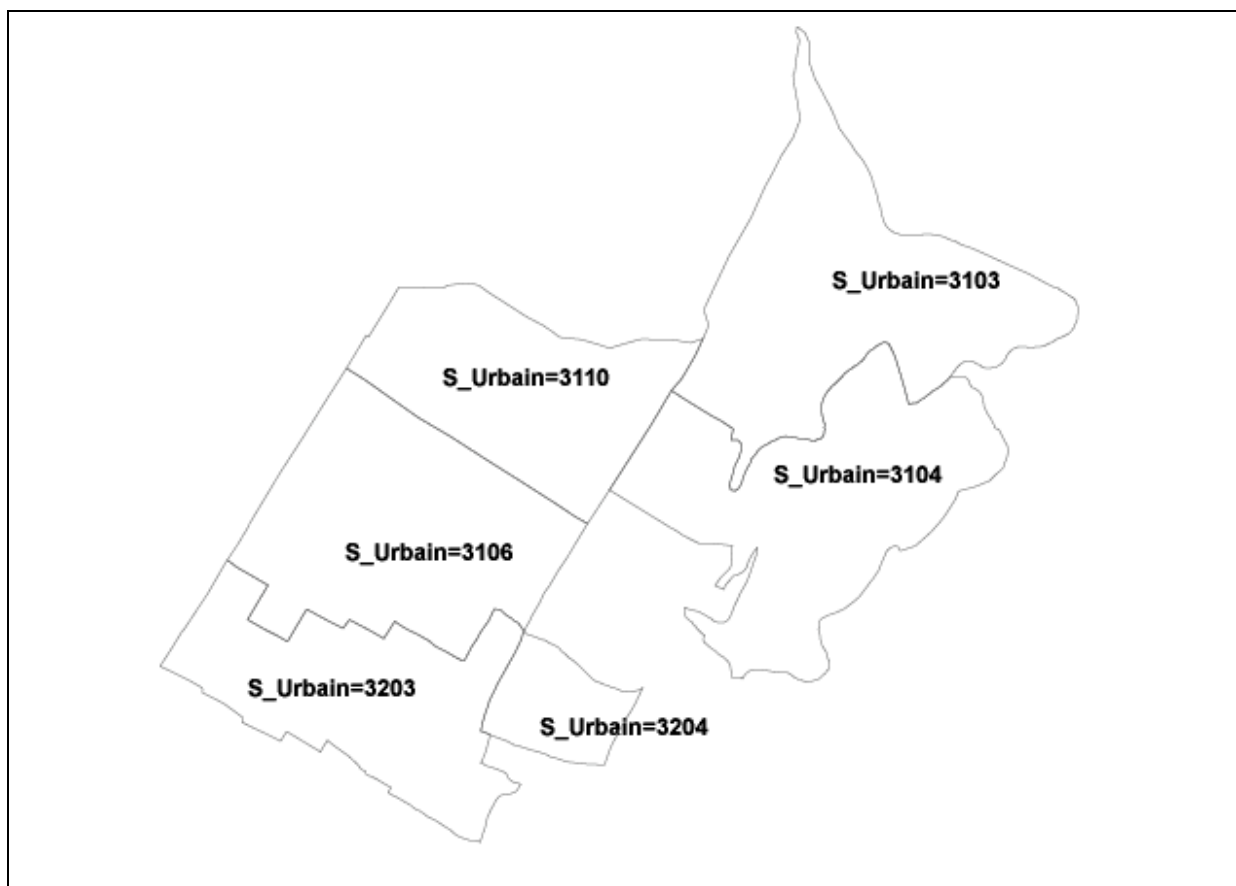
sont celles des recensements généraux de population de Bogotá des années 1973 et 1993 réalisés par le DANE (Département National de Statistique de la Colombie). Dans le cas de l'arrondissement « La Candelaria », la population décroît en passant de 9442 couples de ménages-logements en 1973 à 6414 en 1993. Dans le cas de l'arrondissement « Santa Fé », la population croît légèrement et passe de 22145 couples de ménages-logements en 1973 à 26198 ménages en 1993. Les fichiers d'entrée sont des fichiers de texte où chaque ligne décrit chaque ménage et son logement. Il n'y a pas de données manquantes. Les caractéristiques et les modalités considérées pour décrire les ménages et les logements sont présentées dans l'Annexe I.



Carte 2. Localisation des arrondissements « La Candelaria » et « Santa Fé » du centre historique de Bogotá.  
Source : DANE

### 11.2.1.2. Cartes de secteurs urbains

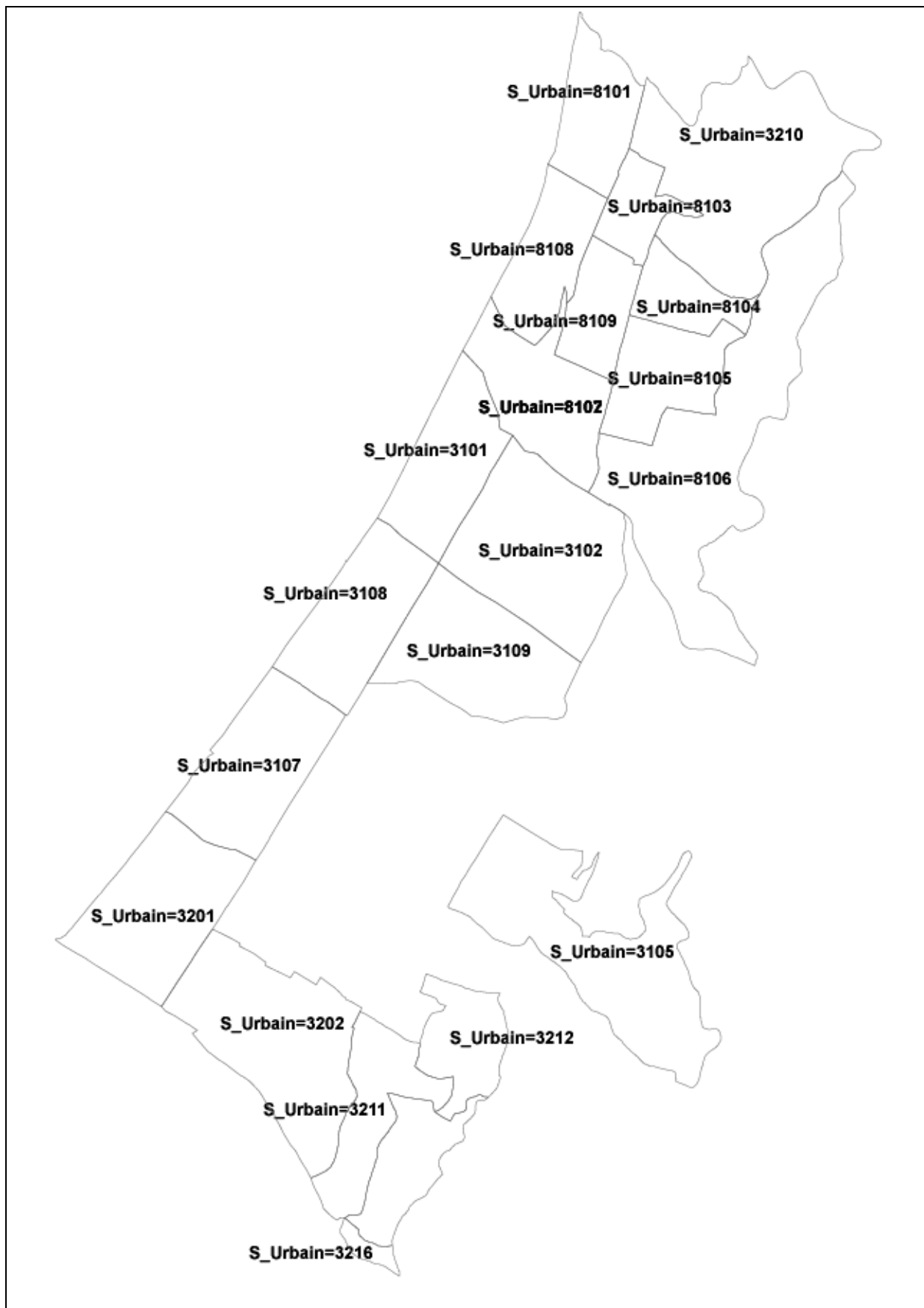
Pour pouvoir produire les cartes de résultat, on a besoin des cartes d'entrée en format *SVG*. Ces cartes contiennent les périmètres des différents secteurs urbains considérés. Il est courant de trouver des fonds de cartes sous le format *shp* (shapefile<sup>20</sup>), il est donc nécessaire de transformer ces cartes à l'aide d'un *parseur*. Il existe différents *parseurs*, dont certains sont gratuits. Nous avons utilisé le parseur libre *shp2svg*<sup>21</sup>. Afin de pouvoir identifier de façon plus aisée les différents secteurs urbains, nous avons ajouté certaines informations à chaque valise définissant le périmètre du secteur. Ce format *SVG* modifié est défini par la DTD présentée dans le chapitre précédent (cf. section 10.2.1.). Les deux cartes des arrondissements « La Candelaria » et « Santa Fé » sont présentées ci-après.



Carte 3. Les secteurs urbains (secteurs des recensements de 1973 et 1993) de l'arrondissement « La Candelaria » à Bogotá. Source : DANE

<sup>20</sup> shapefile est un format de fichiers qui représentent des couches graphiques dans un SIG (Système d'Information Graphique). Initialement développé par la société ESRI il est lisible par leur logiciel commercial ARCGIS (cf. <http://www.esri.com/>), mais aussi par des logiciels libres tels que MapServer (cf. <http://mapserver.gis.umn.edu/>) et Grass (<http://grass.itc.it/>).

<sup>21</sup> cf. <http://www.carto.net/papers/svg/utis/shp2svg/>



Carte 4. Les secteur urbains (secteurs des recensements de 1973 et 1993) de l'arrondissement « Santa Fé » à Bogotá. Source : DANE

### 11.2.2. Prétraitement des données

A partir des fichiers de texte qui contiennent la description sous forme de caractéristiques et de modalités des populations initiale et finale, nous produisons trois fichiers utilisés dans les simulations :

- un fichier de texte qui contient la description sous forme de profils des ménages et logements. Nous appelons ce fichier *fichier de profils*.
- les fichiers *HouseholdsMapping.xml*,
- les fichiers *HousingsMapping.xml* (cf. section 10.2.2.)

Ces deux derniers types de fichiers permettent de décoder les profils de chaque ménage et de chaque logement. Le fichier de profils est divisé en deux parties : l'entête et les données. L'entête contient trois lignes : la première ligne contient le nombre de variables et le nombre de modalités qui décrivent les ménages ; la deuxième contient le nombre de variables et le nombre de modalités qui décrivent les logements ; la troisième ligne contient le nombre de données dans la partie des données. La partie des données contient la description sous forme de profils des couples ménages-logements. Chaque ligne dans cette partie contient la description d'un couple ménage-logement. Il existe trois possibilités :

- Les ménages avec un logement sont décrits par une ligne contenant d'abord le profil du ménage, ensuite le profil du logement et finalement le secteur urbain où le logement est localisé.
- Les ménages sans logements sont décrits par une ligne contenant le profil du ménage suivi de deux '\*'.
- Les logements sans ménages sont décrits par une ligne qui commence par un '\*' suivi du profil du logement et du secteur urbain du logement.

Le secteur urbain du logement est représenté par une chaîne de caractères (le nom du secteur). Un profil de ménage ou de logement est un chiffre entier positif dont la représentation binaire correspond à la description sous forme disjontive complète du ménage ou du logement représenté (cf. Annexe II, Représentation des profils).

Des commentaires sont générés automatiquement afin d'expliquer les informations principales. Les lignes de commentaires commencent par la chaîne de caractères « // ». Dans l'encadré ci-dessous, on présente un extrait d'un fichier de profils.

```

//AUTOMATIC GENERATED FILE
// * MEANS NO PRESENT VALUE
// DISTRIBUTION OF HOUSEHOLDS (FIRST COLUMN) AND HOUSINGS (SECOND COLUMN)
// AND URBAN SECTEURS (THIRD COLUMN)
// 11 VARIABLES WITH 43 MODALITIES THAT DESCRIBE THE HOUSEHOLDS
11      43
// 11 VARIABLES WITH 29 MODALITIES THAT DESCRIBE THE HOUSINGS
11      29
// 26198 PAIRS HOUSEHOLDS:HOUSINGS
26198
// THE DATA :
4829697090176      281090390      S_URBAN=8101
2630686482948      46406230      S_URBAN=8101
*      46209366      S_URBAN=8101
4831843615233      281090390      S_URBAN=8101
4828887679489      281090390      S_URBAN=8101
*      281090390      S_URBAN=8101
4828085518976      281090390      S_URBAN=8101
4828627607872      155327061      S_URBAN=8101
2560892216328      155327061      S_URBAN=8101
5111021708289      155327061      S_URBAN=8101
2911997403394      146938453      S_URBAN=8101
2912006775044      146938453      S_URBAN=8101
2560908996609      *      S_URBAN=8101
4829709738272      155327062      S_URBAN=8101
4829707903496      281090390      S_URBAN=8101
5111079404033      281090390      S_URBAN=8101
2630677045632      *      S_URBAN=8101
4831843591296      281094486      S_URBAN=8101
4763388317697      281090390      S_URBAN=8101
5042359927072      281094486      S_URBAN=8101
4829724689024      289679702      S_URBAN=8101
5042310578560      281094486      S_URBAN=8101
2630936305922      282073430      S_URBAN=8101
4760988434816      281155926      S_URBAN=8101
4828652710016      281090390      S_URBAN=8101
4763188076832      46209366      S_URBAN=8101
4763186245640      281155926      S_URBAN=8101
4828651659780      281549142      S_URBAN=8101
4832108904992      281090390      S_URBAN=8101
5111021717505      146938453      S_URBAN=8101
4831857181192      46209382      S_URBAN=8101
5111029056001      54598246      S_URBAN=8101
2630702211332      281286998      S_URBAN=8101
4759908090128      147069525      S_URBAN=8101
4828626584608      46209366      S_URBAN=8101
4829961396744      47192406      S_URBAN=8101
4760981898368      46209366      S_URBAN=8101
4829726540289      46213462      S_URBAN=8101
5042309579009      46209366      S_URBAN=8101
4829700325648      47192406      S_URBAN=8101
4759907107856      46209366      S_URBAN=8101
5111029072385      289483094      S_URBAN=8101
4829726474753      281094486      S_URBAN=8101
4760988426560      146942550      S_URBAN=8101
2630703261704      47196502      S_URBAN=8101
5111046832656      47192406      S_URBAN=8101

```

Encadré 24. Extrait d'un fichier de profils.

### 11.2.3. Paramétrage des simulations

#### 11.2.3.1. Paramètres de mobilité (voir 11.1.)

##### 11.2.3.1.1. Stratégies de mobilité

Les secteurs urbains considérés dans les simulations sont les secteurs des recensements (divisions spatiales plus petites que les arrondissements) de 1973 et 1993. Les cartes présentées dans la section 11.2.1.2. montrent les différents secteurs des recensements pour chacun des deux arrondissements considérés. Nous avons considéré trois définitions des coûts de déménager entre ces secteurs selon trois hypothèses différentes :

Cas 1 : Les ménages préfèrent déménager à l'intérieur de leur secteur urbain courant. Le coût de déménager dans le même secteur urbain est de 0. Etant donné que les zones considérées sont relativement petites par rapport à la ville, nous avons considéré un coût de déménager de 0,5 pour des déménagements entre toutes les paires de secteurs urbains différents.

Cas 2 : Les ménages préfèrent déménager vers des secteurs différents de leur secteur urbain courant. Nous avons donc considéré que le coût de déménager entre paires de secteurs urbains différents est de 0 et que le coût pour déménager à l'intérieur du même secteur urbain est 0,5.

Cas 3 : Absence de préférence. Les coûts de déménagement entre les paires de secteurs urbains ont été définis de manière aléatoire. Nous avons utilisé une distribution normale centrée dans la valeur 0,5 pour définir ces coûts aléatoires.

##### 11.2.3.1.2. Evolution de l'urgence pour résoudre la demande

Le facteur d'augmentation du prix (*facteurPrix*) proposé lors d'une enchère par les groupes-sociaux-non-localisés a été défini dans la section 6.2.2.2.2. Ce facteur représente l'urgence qu'un groupe-social-localisé a pour résoudre sa demande courante de logements. Dans cette définition, le *facteurPrix* augmente de 1% ( $k = 1,01$  dans l'équation de *facteurPrix*, cf. section 6.2.2.2.2.) sur chaque recherche d'enchère effectuée par un groupe-social-non-localisé. En plus de cette situation, nous avons considéré les deux cas d'évolution du *facteurPrix* suivants :

- Dans le premier cas nous considérons une réduction rapide du facteur : diminution de 50% sur chaque recherche ( $k=0,5$  dans l'équation de *facteurPrix*).
- Dans le deuxième cas nous avons considéré une évolution rapide du facteur : évolution de 100% sur chaque recherche ( $k=2,0$  dans l'équation de *facteurPrix*).

### 11.2.3.2. Paramètres de formation des groupes

Dans le mécanisme de formation des groupes, nous avons considéré trois algorithmes de classification différents : l'algorithme amélioré de cartes adaptatives de Kohonen (SOM), présenté dans la section 8.2, l'algorithme basé sur les nuages de particules (PART), présenté dans la section 8.4 et l'algorithme Multi-Kmeans présenté dans la section 8.5).

Nous nous intéressons principalement à la définition du nombre de classes trouvées dans les classifications. Une analyse plus approfondie des autres paramètres dans chaque algorithme de classification est hors du propos de notre travail courant. Nous nous intéressons au nombre de classes car il est lié au niveau d'abstraction utilisé dans le mécanisme de formation de groupes.

En effet, plus le nombre de classes considéré est grand, plus les groupes obtenus sont proches du niveau microscopique ; au contraire, plus le nombre de classes considéré est petit, plus le niveau d'abstraction des groupes est élevé.

Dans un premier temps, dans le choix du paramétrage des différents algorithmes, le critère fondamental est de trouver un nombre maximal de 25 classes pour les ménages et pour les logements. Nous proposons ainsi un niveau élevé d'abstraction. Dans un deuxième temps nous considérons un nombre plus important de classes afin de tester la sensibilité des résultats des simulations lorsqu'on diminue le niveau d'abstraction lors de la formation des groupes.

#### 11.2.3.2.1. Niveau d'abstraction élevé

Le paramétrage des algorithmes de classification initialement considérés est le suivant :

##### *11.2.3.2.1.1. Paramétrage de la formation des groupes par l'algorithme SOM*

Pour l'algorithme SOM, nous avons considéré les paramètres présentés ci-après. Le paramétrage est le même pour les classifications des logements et des ménages.

- Configuration de la grille : Nous avons utilisé une grille rectangulaire de 5x5 ( $nxgrille=5$  ;  $nygrille=5$ ) neurones dans la couche cachée. Ce qui donne la possibilité de trouver maximum 25 classes.
- Paramètre initial d'actualisation des poids des neurones ( $initTau$ ) : 0,02.
- Nombre maximal d'actualisations des poids de neurones ( $maxIterSOM$ ) : 100
- Nombre d'exécutions de l'algorithme ( $maxIterOptimisation$ ) pour choisir la meilleure partition (selon le critère de Davis-Bouldin) : 30.

Les valeurs des trois derniers paramètres ont été choisies de manière arbitraire.

##### *11.2.3.2.1.2. Paramétrage de la formation des groupes par l'algorithme Multi-KMeans*

Pour l'algorithme Multi-Kmeans, nous avons considéré les paramètres présentés ci-après. Le paramétrage est le même pour les classifications des logements et des ménages.

- Nombre minimal de classes (*minK*) : 2.
- Nombre maximal de classes (*maxK*) : 25
- Nombre d'actualisation des centres pour un même nombre de classes (*maxIterKmeans*) : 100. La valeur de ce paramètre a été choisie de manière arbitraire.

#### *11.2.3.2.1.3 Paramétrage de la formation des groupes par l'algorithme basé sur les nuages de particules (PART)*

Pour l'algorithme PART, nous avons considéré les paramètres présentés ci-après. Le paramétrage est le même pour les classifications des logements et des ménages.

- Détection des classes avec l'algorithme Multi-Kmeans : nombre minimal de classes (*minK*) : 2 ; nombre maximal de classes (*maxK*) : 25 ; nombre d'actualisations des centres (*maxIterKmeans*) : 100. La valeur de ce dernier paramètre a été choisie de manière arbitraire.
- Actualisation des localisations des particules : nombre maximal d'actualisation des localisations (*maxIterations*) : 100 ; énergie minimale (*min\_énergie*) : 0,001. Les valeurs de ces deux paramètres ont été choisies de manière arbitraire.

#### 11.2.3.2.2. Niveau d'abstraction faible

Pour étudier les effets de la réduction du niveau d'abstraction dans les simulations lors de la formation des groupes, nous avons considéré un nombre relativement grand de classes dans la classification des ménages et des logements.

Pour cet effet, nous avons utilisé uniquement l'algorithme PART avec le paramétrage suivant :

- Détection des classes avec l'algorithme Multi-Kmeans : intervalle pour le nombre de classes de ménages : [*minK*= 70; *maxK*= 100] ; intervalle pour le nombre de classes de logements [*minK*= 20; *maxK*=50]; nombre d'actualisations des centres (*maxIterKmeans*) : 100. La valeur de ce dernier paramètre a été choisie de manière arbitraire.
- Actualisation des localisations des particules : nombre maximal d'actualisation des localisations (*maxIterations*) : 100 ; énergie minimale (*min\_énergie*) : 0,001. Les valeurs de ces deux paramètres ont été choisies de manière arbitraire.

#### 11.2.3.3. Paramètres de construction des règles d'évolution

Pour construire les règles d'évolution de la population dans le cas du modèle synthétique, nous devons d'abord déterminer les matrices de distribution des populations initiale et finale



(cf. section 7.2.3.5.). Ces matrices sont construites à partir du croisement des groupes sociaux et des types d'habitat obtenus par la classification des populations et des parcs du logement réels pour les années initiale et finale. Le nombre de classes considérées pour constituer ces groupes sociaux et ces types d'habitat définit le niveau d'abstraction des règles d'évolution.

Nous avons considéré deux niveaux d'abstraction différents pour la construction de ces règles :

- Dans un premier temps, nous considérons un niveau d'abstraction faible (proche du niveau microscopique). Dans ce cas, les groupes sociaux et types d'habitat ont été créés en utilisant l'algorithme de classification *ClusteringBasique*.
- Dans un deuxième temps, nous avons augmenté le niveau d'abstraction, en utilisant l'algorithme de classification *SOM* avec un nombre réduit de classes pour la construction de groupes sociaux et de types d'habitat nécessaires dans la construction des règles d'évolution.

#### 11.2.3.3.1 Niveau d'abstraction faible

Dans le cas de la construction des règles d'évolution à partir des classifications avec l'algorithme de *ClusteringBasique*, nous traitons directement les profils réels de ménages et des logements.

Les règles d'évolution qui en résultent permettent la reconstruction « exacte » des profils et nombre de ménages et de logements pour l'année de simulation finale. Dans le tableau ci-dessous, nous présentons quelques statistiques du nombre de règles trouvées pour chacun des arrondissements étudiés :

	La Candelaria	Santa Fé
Nombre de règles de suppression de ménages (nombre de profils de ménages pour l'année 1973)	6052	11152
Nombre de règles de création des ménages (nombre de profils de ménages pour l'année 1993)	4577	13045
Nombre de règles de suppression des logements (nombre de profils de logements pour l'année 1973)	398	604
Nombre de règles de création des logements (nombre de profils de logements pour l'année 1993)	640	983

Tab 8. Nombre de règles d'évolution pour les arrondissements « La Candelaria » et « Santa Fé » construites à l'aide de l'algorithme de classification *ClusteringBasique*. Chaque règle correspond à un profil exacte de ménage ou de logement.

#### 11.2.3.3.2 Niveau d'abstraction élevé

Dans le cas de la construction des règles d'évolution à partir des classifications avec l'algorithme *SOM*, nous traitons les profils synthétiques. Chaque profil synthétique représente

le profil moyen des micro-entités (des ménages ou des logements, selon le cas) appartenant à une même classe. Le niveau d'abstraction qui dépend du nombre de classes trouvées est relativement élevé. Pour construire les règles d'évolution à partir des classifications avec l'algorithme SOM, nous avons considéré le paramétrage suivant (le paramétrage est le même pour les classifications des ménages et pour les classifications des logements) :

- Configuration de la grille : Nous avons utilisé une grille rectangulaire de 10x10 ( $nxgrille=10$  ;  $nygrille=10$ ) neurones dans la couche cachée, ce qui donne la possibilité de trouver maximum 100 classes.
- Paramètre initial d'actualisation des neurones ( $initTau$ ) : 0,02.
- Nombre maximal d'actualisations des poids de neurones ( $maxIterSOM$ ) : 100
- Nombre d'exécutions de l'algorithme ( $maxIterOptimisation$ ) pour choisir la meilleure partition (selon le critère de Davis-Bouldin) : 30.

Le nombre de règles d'évolution trouvées avec cet algorithme est présenté dans le tableau ci-dessous :

	La Candelaria	Santa Fé
Nombre de règles de suppression de ménages (nombre de profils de ménages pour l'année 1973)	83	76
Nombre de règles de création des ménages (nombre de profils de ménages pour l'année 1993)	82	71
Nombre de règles de suppression des logements (nombre de profils de logements pour l'année 1973)	17	20
Nombre de règles de création des logements (nombre de profils de logements pour l'année 1993)	32	15

Tab 9. Nombre de règles d'évolution pour les arrondissements « La Candelaria » et « Santa Fé » construites à partir de l'algorithme de classification SOM avec une grille rectangulaire de 10x10. Chaque règle correspond à un profil moyen d'une classe de ménages ou d'une classe de logements.

#### 11.2.3.4. Paramétrage de référence

Nous avons défini un paramétrage de référence pour les simulations qui nous sert de point de comparaison pour mesurer la sensibilité des résultats par rapport aux variations des différents paramètres. Ce paramétrage est le suivant :

- Coûts de déménager : nous avons considéré l'hypothèse selon laquelle les ménages préfèrent déménager dans leur secteur courant (cas de figure 1 dans la section 11.2.3.1.).

- Evolution du facteur de prix pour les groupes-sociaux-non-localisés : nous avons considéré une évolution de 1% sur chaque recherche d'enchères ( $k = 1,01$  dans l'équation du *facteurPrix*, cf. section 6.2.2.2.2.).
- Nombre maximum de classes considérées pour la formation des groupes : nous considérons un niveau d'abstraction élevé avec un maximum de 25 classes pour les trois algorithmes de classification considérés (cf. section 11.2.3.2.1.).
- Algorithme de classification utilisé dans la construction des règles d'évolution : Comme point de référence, nous avons considéré l'algorithme de *ClusteringBasique* (cf. section 11.2.3.3.1.), ce qui donne un niveau d'abstraction faible dans la construction des règles d'évolution.

#### **11.2.4. Description des types de simulations effectuées**

Nous avons réalisé différents types de simulations sur les données des arrondissements « Santa Fé » et « La Candelaria » (cf. section 11.2.1.1.). Les types de simulations sont définis selon le modèle (synthétique, étendu avec et sans intervention des ménages dans la prise de décision lors du déménagement) et l'algorithme de classification utilisé dans la formation des groupes. Initialement, nous avons exécuté ces simulations avec le paramétrage de référence pour obtenir des points de comparaison quant à la qualité des simulations. Ensuite, nous avons changé graduellement les valeurs des paramètres afin de pouvoir mesurer la stabilité des résultats par rapport au paramétrage et trouver le meilleur paramétrage. Toutes les simulations ont été réalisées sur 20 années de simulation entre les années 1973 et 1993 pour les deux arrondissements considérés.

##### 11.2.4.1. Simulations avec le modèle synthétique

Nous avons réalisé plusieurs simulations avec le modèle synthétique pour chacun des jeux de données. Les matrices de distribution initiale et finale ont été construites à partir des classifications conjointes des données initiales et finales. Autrement dit, on a considéré l'ensemble des populations initiales et finales et l'on a déterminé des groupes sociaux et des types d'habitat. La classification conjointe permet de prendre en considération les mêmes ensembles de groupes sociaux et de types d'habitat pour les populations initiale et finale. Les matrices d'évolution des groupes sociaux et des types d'habitat sont construites à partir des matrices de distribution initiale et finale. Pour la classification conjointe, nous avons considéré les trois algorithmes et le paramétrage considérés dans la section 11.2.3.2.1.

### 11.2.4.2. Simulations avec le modèle étendu

#### 11.2.4.2.1. Localisation du processus de décision

Avec le modèle étendu, nous avons réalisé deux types de simulation par rapport à la localisation de la décision de déménager. Dans le premier type de simulation, nous avons utilisé une instance de la classe *CDMBasique* en tant que contrôleur de la distribution microscopique de population. Cette classe ne permet pas la participation des ménages au processus de décision. On est donc dans le cas d'un processus de décision localisé uniquement au niveau mésoscopique (cf. section 7.2.2.1.). Dans le deuxième type de simulation, nous avons utilisé une instance de la classe *CDMInterventionIndividuelle* en tant que contrôleur de la distribution microscopique de la population. Dans ce cas, les ménages ont une participation « faible » à la prise de la décision de déménager (cf. section 7.2.2.2).

#### 11.2.4.2.2. Paramétrage initial

Nous avons réalisé plusieurs simulations avec le modèle étendu sur les deux jeux de données en changeant à chaque fois l'algorithme de classification utilisé dans la formation des groupes. Nous avons utilisé trois algorithmes de classification différents pour la formation des groupes sociaux et des types d'habitat : SOM, PART et Multi-Kmeans. Le paramétrage de ces algorithmes est initialement celui défini dans la section 11.2.3.2.1., c'est-à-dire en considérant un niveau d'abstraction élevé dans la mécanisme de formation de groupes. D'autre part, on a considéré un niveau faible dans la construction des règles d'évolution, comme il a été décrit dans la section 11.2.3.3.1.

### 11.2.4.3. Recapitulatif des différents types de simulations effectués

Dans le tableau ci-dessous, nous présentons un recapitulatif des différents types de simulations effectuées. La colonne « code » est utilisée dans la suite du document pour faire référence à chaque type de simulation.

Modèle utilisé	Code	Description	
Modèle synthétique	MS-SOM	Création des matrices de distribution finale et initiale en utilisant l'algorithme SOM	
	MS-MKM	Création des matrices de distribution finale et initiale en utilisant l'algorithme Multi-Kmeans	
	MS-PART	Création des matrices de distribution finale et initiale en utilisant l'algorithme PART	
Modèle Etendu	ME-DM-SOM	Formation des groupes avec l'algorithme SOM	Prise de décision de déménager au niveau

	ME-DM-MKM	Formation des groupes avec l'algorithme Multi-Kmeans	mésoscopique
	ME-DM-PART	Formation des groupes avec l'algorithme PART	
	ME-DMI-SOM	Formation des groupes avec l'algorithme SOM	Prise de décision de déménager au niveau mésoscopique avec faible intervention du ménage
	ME-DMI-MKM	Formation des groupes avec l'algorithme Multi-Kmeans	
	ME-DMI-PART	Formation des groupes avec l'algorithme PART	

Tab 10. Tableau recapitulatif des 9 types de simulations effectuées.

## 11.3 Résultats des expériences

### 11.3.1. Résultats préliminaires

#### *11.3.1.1. Stabilité du mécanisme de mobilité et précision des résultats*

Afin de mesurer la stabilité du mécanisme de mobilité, nous avons effectué chaque type de simulation 30 fois sur les jeux de données des arrondissements considérés (« Santa Fé » et « Candelaria »). Dans chaque exécution des simulations, nous avons utilisé le paramétrage de référence présenté dans la section 11.2.3.4. Nous avons mesuré la stabilité du coefficient de corrélation pour chaque type de simulation. Dans le cas des simulations étendues, afin de ne pas influencer le calcul de la stabilité du mécanisme de mobilité par la variabilité des processus de classification, nous avons d'abord précalculé les populations de chaque année de simulation en utilisant les règles d'évolution et ensuite nous avons prédéterminé les groupes sociaux et types d'habitat pour chaque année de simulation avec chacun des 3 algorithmes de classification considérés pour la formation de groupes (SOM, PART et Multi-Kmeans). Les groupes sociaux et types d'habitat ainsi prédéterminés ont été utilisés pour chacune des 30 exécutions de chaque type de simulation.

Pour chaque exécution de chaque type de simulation, on a calculé le coefficient de corrélation entre la matrice de distribution finale simulée et la matrice de distribution finale réelle. Sur chaque ensemble des 30 mesures, nous avons calculé la moyenne et l'écart type de la valeur du coefficient de corrélation. Dans les tableaux ci-dessous nous présentons ces

résultats. On observe que l'écart type du coefficient de corrélation sur tous les types de simulation effectués est relativement petit par rapport au coefficient de corrélation. La conclusion qu'on peut en tirer est que la variabilité du coefficient de corrélation est peu significative et que le mécanisme de mobilité est donc très stable. Dans les sections suivantes, nous présentons une analyse de ces premiers résultats d'abord par rapport à l'algorithme de classification utilisé dans le mécanisme de formation des groupes et ensuite par rapport au modèle utilisé.

	La Candelaria		Santa Fé	
	ZNCC	Ecart type du ZNCC	ZNCC	Ecart type du ZNCC
MS-SOM	0,949	0,007	0,984	0,002
MS-MKM	0,524	0,017	0,722	0,011
MS-PART	0,455	0,024	0,809	0,012

Tab 11. Coefficients de corrélation et écart type pour trois types de simulations synthétiques avec le paramétrage de référence. Chaque type de simulation a été exécuté 30 fois sur les données des arrondissements « La Candelaria » et « Santa Fé »

	La Candelaria		Santa Fé	
	ZNCC	Ecart type du ZNCC	ZNCC	Ecart type du ZNCC
ME-DM-SOM	0,240	0,011	0,275	0,007
ME-DM-MKM	0,467	0,032	0,495	0,008
ME-DM-PART	0,812	0,014	0,920	0,010

Tab 12. Coefficients de corrélation et écart type pour des simulations avec le modèle ME-DM et trois algorithmes de formation de groupes différents et avec le paramétrage de référence. Chaque type de simulation a été exécuté 30 fois sur les données des arrondissements « La Candelaria » et « Santa Fé »

	La Candelaria		Santa Fé	
	ZNCC	Ecart type du ZNCC	ZNCC	Ecart type du ZNCC
ME-DMI-SOM	0,693	0,005	0,360	0,004
ME-DMI-MKM	0,265	0,017	0,422	0,021
ME-DMI-PART	0,847	0,003	0,932	0,007

Tab 13. Coefficients de corrélation et écart type pour des simulations avec le modèle ME-DMI et trois algorithmes de formation de groupes différents et avec le paramétrage de référence. Chaque type de simulation a été exécuté 30 fois sur les données des arrondissements « La Candelaria » et « Santa Fé »

11.3.1.2. Sensibilité des résultats par rapport à l'algorithme de classification utilisé dans la formation de groupes

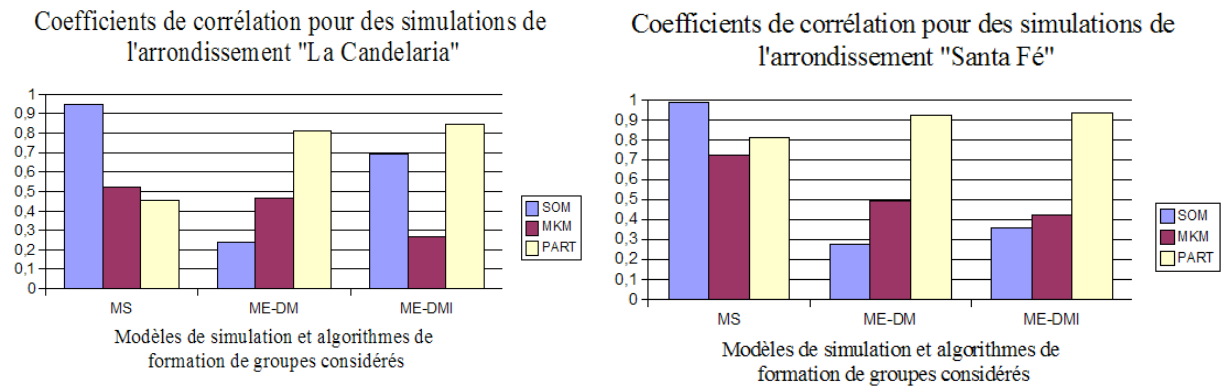


Figure 62. Coefficients de corrélation pour les 9 types de simulation avec le paramétrage de référence.

Dans les graphiques ci-dessus, on présente les valeurs du coefficient de corrélation pour les neuf types de simulation réalisées avec le paramétrage de référence. Dans ces graphiques, on peut constater un phénomène curieux : dans le cas du modèle synthétique, on obtient de meilleurs résultats (plus grandes valeurs du coefficient de corrélation) lorsqu'on utilise l'algorithme de classification SOM pour déterminer les groupes sociaux et les types d'habitat. Par contre, lorsqu'on exécute des simulations avec les modèles étendus, les simulations qui utilisent l'algorithme SOM présentent une très basse performance. Les meilleurs résultats sont, dans ce cas-là, obtenus avec les simulations qui utilisent l'algorithme PART pour construire les groupes sociaux et les types d'habitat.

Pour essayer de comprendre la basse performance des simulations des types ME-MD-SOM, ME-MDI-SOM, ME-MD-MKM et ME-MDI-MKM, nous analysons les résultats des différentes classifications pour l'année finale de simulation et pour l'année de validation (année finale réelle). Dans cette analyse, il faut tenir compte qu'étant donné que les règles d'évolution ont été construites avec l'algorithme de *ClusteringBasique*, les profils et nombre des micro-entités (pour les ménages et pour les logements respectivement) sont exactement les mêmes pour l'année finale de simulation et pour l'année de validation (cf. section 11.2.3.5.1). Autrement dit, les ensembles de profils qui représentent les ménages et les logements pour l'année finale de simulation et pour l'année de validation sont exactement les mêmes<sup>22</sup>. Dans ces conditions, avec une même méthode de classification, on devrait s'attendre à trouver les mêmes partitions de ménages et de logements pour les deux années.

<sup>22</sup> Il faut noter que même si les populations et les parcs de logement sont les mêmes pour l'année finale de simulation et pour l'année de validation, la distribution spatiale des groupes sociaux, n'est pas forcément la même pour ces deux années.

Deux partitions identiques doivent avoir le même nombre de classes et la même valeur du coefficient de Davis-Bouldin.

Dans les tableaux ci-dessous, nous présentons le nombre de classes et le coefficient de Davis-Bouldin pour les différentes classifications des ménages et des logements. A partir des résultats présentés dans ces tableaux on peut conclure :

- Lorsqu'on utilise l'algorithme basé sur les nuages de particules (PART) pour former les groupes sociaux et les types d'habitat, on obtient les mêmes nombres de classes et les mêmes valeurs du coefficient de Davis-Bouldin pour les données de l'année finale de simulation et de l'année de validation. Dans ces cas-là, on peut donc considérer que les partitions de ménages et de logements sont identiques pour les deux années.
- Lorsqu'on utilise l'algorithme SOM pour former les groupes sociaux et les types d'habitat, les partitions obtenues pour l'année finale de simulation sont différentes de celles obtenues pour l'année de validation, puisque les nombres de classes et les valeurs des coefficients de Davis-Bouldin sont différents pour les deux années.
- Lorsqu'on utilise l'algorithme Multi-Kmeans pour former les groupes sociaux et les types d'habitat, dans le cas de l'arrondissement « La Candelaria », les partitions obtenues pour l'année finale de simulation sont différentes de celles obtenues pour l'année de validation. Dans le cas de l'arrondissement « Santa Fé », même si les nombres de classes trouvées pour l'année finale de simulation et pour l'année de validation sont les mêmes, les coefficients de Davis-Bouldin sont différents ; les partitions ne sont pas identiques.

	Année finale de simulation				Année réelle de validation			
	Groupes Sociaux		Types d'habitats		Groupes Sociaux		Types d'habitats	
	# classes	Davis-Bouldin	# classes	Davis-Bouldin	# classes	Davis-Bouldin	# classes	Davis-Bouldin
SOM	22	3,153	11	2,176	19	2,954	9	2,3522
Multi-KMeans	25	3,318	23	1,968	18	3,505	25	2,056
Particules	7	3,226	12	3,339	7	3,226	12	3,339

Tab 14. Nombre de classes et coefficient de Davis-Bouldin pour les classifications réalisées sur les données de l'année finale simulée et sur les données de l'année de validation pour l'arrondissement « La Candelaria »

	Année finale de simulation	Année réelle de validation
--	----------------------------	----------------------------



	Groupes Sociaux		Types d'habitats		Groupes Sociaux		Types d'habitats	
	# classes	Davis-Bouldin	# classes	Davis-Bouldin	# classes	Davis-Bouldin	# classes	Davis-Bouldin
SOM	23	3,227	10	1,861	20	3,300	8	1,841
Multi-Kmeans	11	3,942	19	1,773	11	3,970	19	1,932
Particules	7	3,425	3	3,183	7	3,425	3	3,183

Tab 15. Nombre de classes et coefficient de Davis-Bouldin pour les classifications réalisées sur les données de l'année finale simulée et sur les données de l'année de validation pour l'arrondissement « Santa Fé »

A partir de ces constats, on peut supposer que les résultats des simulations ont été biaisés par la variabilité due aux algorithmes de classification. Pour prouver cette hypothèse, nous avons effectué les mêmes simulations, mais cette fois-ci, pour les algorithmes Multi-Kmeans et SOM. Nous avons utilisé les mêmes partitions de ménages et les mêmes partitions de logements dans la constitution des distributions de populations finales simulée et réelle (année de validation).

Pour le cas de l'algorithme PART, on a conservé la situation originale. Les résultats des simulations avec ces nouvelles conditions sont présentés dans les tableaux ci-dessous et dans la Figure 63. On observe dans ces graphiques, une amélioration considérable de la qualité des simulations avec le modèle étendu lorsqu'on utilise les algorithmes Multi-Kmeans et SOM dans la formation de groupes.

L'hypothèse de la dépendance des résultats des simulations à la stabilité des algorithmes de classification pour la formation de groupes est démontrée. Dans la suite, nous allons utiliser ces derniers résultats comme point de comparaison pour l'analyse de la sensibilité des simulations aux variations des paramètres.

	La Candelaria		Santa Fé	
	ZNCC	Ecart type du ZNCC	ZNCC	Ecart type du ZNCC
ME-DM-SOM	0,973	0,002	0,972	0,002
ME-DM-MKM	0,594	0,032	0,831	0,017

Tab 16. Coefficients de corrélation et écart type pour des simulations avec le modèle ME-DM et 2 algorithmes de formation de groupes différents. Les partitions utilisées pour construire les groupes pour l'année finale de simulation et l'année de validation sont les mêmes. Chaque type de simulation a été exécuté 30 fois sur les données des arrondissements « La Candelaria » et « Santa Fé ».

	La Candelaria		Santa Fé	
	ZNCC	Ecart type du ZNCC	ZNCC	Ecart type du ZNCC
ME-DMI-SOM	0,990	0,001	0,973	0,002
ME-DMI-MKM	0,531	0,027	0,829	0,015

Tab 17. Coefficients de corrélation et écart type pour des simulations avec le modèle ME-DMI et 2 algorithmes de formation de groupes différents. Les partitions utilisées pour construire les groupes pour l'année finale de simulation et l'année de validation sont les mêmes. Chaque type de simulation a été exécuté 30 fois sur les données des arrondissements « La Candelaria » et « Santa Fé ».

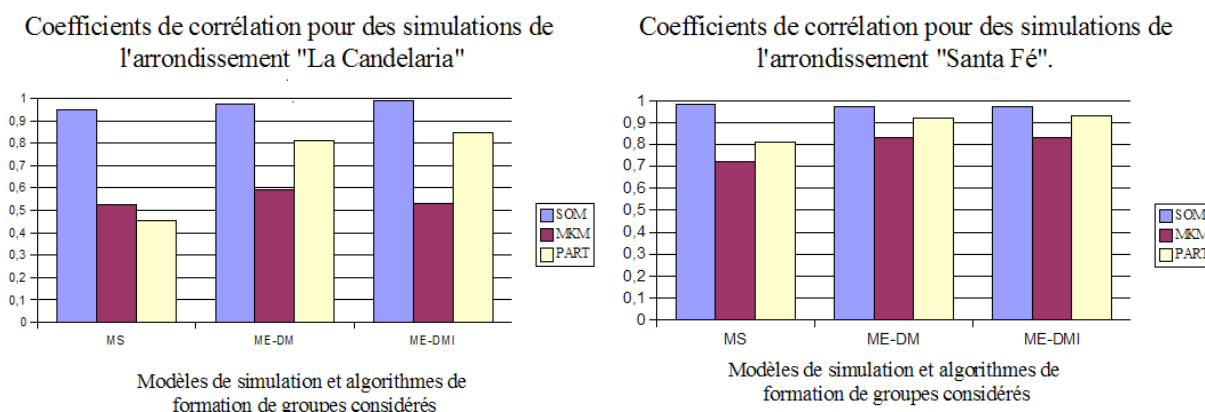


Figure 63. Coefficients de corrélation pour les différents modèles et algorithmes de formation de groupes considérés. Avec les mêmes partitions pour les ménages et les logements des années finale simulée et de validation (année finale réelle).

### 11.3.1.3. Choix du modèle et de l'algorithme de classification

Dans la figure ci-dessus, on observe que, de manière générale, les meilleures valeurs du coefficient de corrélation sont atteintes par le modèle étendu. D'un autre côté, on n'observe pas une différence significative entre les simulations avec et sans participation des ménages au processus de décision du déménagement.

D'un autre côté, on observe aussi (cf. Figure 63), une forte dépendance du coefficient de corrélation à l'algorithme de classification utilisé dans la constitution des groupes sociaux et des types d'habitat. De manière générale, les meilleurs résultats (coefficients de corrélations les plus élevés) sont obtenus avec l'algorithme de classification *SOM*, en deuxième position se trouvent les simulations avec l'algorithme de classification *PART* et en dernière position se trouvent les simulations effectuées avec l'algorithme de classification *Multi-KMeans*. D'autre part, dans les tableaux 14 et 15, dans la section précédente, on observe que pour le cas du modèle étendu, les meilleurs coefficients de Davis-Bouldin sont obtenus par les classifications avec l'algorithme *SOM*. En deuxième position on trouve les coefficients de Davis-Bouldin des classifications avec l'algorithme *PART*, et en dernière position on trouve les coefficients des classifications avec l'algorithme *Multi-KMeans*. Il semblerait exister une relation entre la

valeur du coefficient de Davis-Bouldin et la valeur du coefficient de corrélation des simulations. Autrement dit, il semblerait exister une corrélation entre la qualité des classifications et la capacité des simulations à reproduire la distribution spatiale des groupes sociaux dans la ville.

Pour vérifier l'existence de cette corrélation, nous présentons ci-dessous un graphique qui croise les différents valeurs du coefficients de Davis-Bouldin avec les valeurs du coefficient de corrélation pour les 9 types de simulations réalisées (cf. section 11.2.4.3.). Pour le cas des simulations avec le modèle étendu, nous représentons graphiquement la valeur du coefficient de Davis-Bouldin pour les partitions dans l'année finale de simulation contre la valeur moyenne du coefficient de corrélation de la simulation. Pour le cas des simulations avec le modèle synthétique, nous représentons graphiquement la valeur du coefficient de Davis-Bouldin de la classification conjointe faite pour constuire les distributions de populations initiale et finale (cf. section 11.2.4.1.) contre la valeur moyenne du coefficient de corrélation de la simulation.

Dans le graphique ci-dessous, on observe deux choses :

- Dans le cas des groupes sociaux, pour la plupart des valeurs de Davis-Bouldin inférieures à 3,5, on obtient des valeurs du coefficient de corrélation supérieures à 0,9 ; tous les points avec des valeurs de Davis-Bouldin supérieures à 3,5 sont liés à des coefficients de corrélation plus bas que 0,9.
- Dans le cas des types d'habitat, il n'existe pas une tendance claire : on observe que les coefficients de corrélation les plus hauts sont liés à des valeurs faibles du coefficient de Davis-Bouldin, mais on ne peut pas dire que pour toute valeur faible du coefficient de Davis-Bouldin, on obtienne des bonnes valeurs du coefficient de corrélation.

A partir de ces constats, on peut conclure que les meilleurs résultats des simulations (les plus grandes valeurs des coefficients de corrélation) sont obtenus lorsque les classifications sont mieux notées (les plus bas coefficients de Davis-Bouldin).

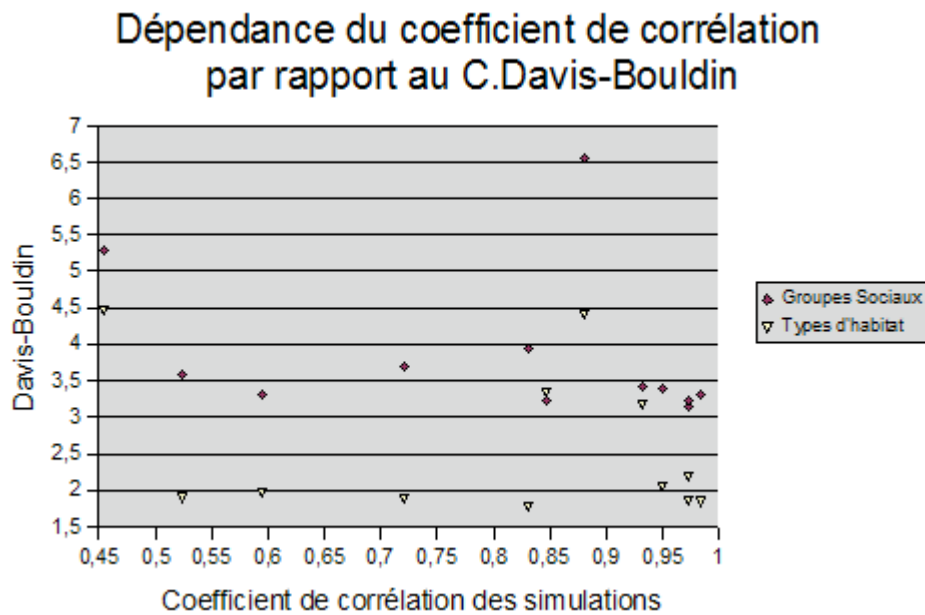


Figure 64. Croissement entre la valeur du coefficient de corrélation des simulations avec la valeur du coefficient de Davis-Bouldin des partitions utilisées dans la construction des groupes sociaux et types d'habitat.

#### 11.3.1.4. Conclusion des résultats préliminaires

A partir des résultats présentés jusqu'à ce point du document, on peut tirer les conclusions suivantes :

- Les meilleurs résultats sont obtenus pour le modèle étendu. Il semble donc nécessaire de considérer le niveau microscopique dans les simulations.
- Les différences entre les modèles étendus ME-MD et ME-MDI ne sont pas significatives, on peut alors conclure qu'il n'est pas important de considérer la participation « faible » des ménages à la prise de décision du déménagement. Le processus de prise de décision placé uniquement au niveau des groupes est suffisant pour obtenir de bons résultats de simulation.
- La qualité des partitions utilisées pour produire les groupes sociaux et les types d'habitat a un impact considérable dans les résultats des simulations.
- La stabilité de l'algorithme de classification utilisé dans la formation des groupes sociaux et des types d'habitat, a une influence majeure dans la qualité des résultats des simulations. L'algorithme de classification le plus stable parmi les trois algorithmes présentés est l'algorithme basé sur les nuages des particules (PART).

A partir des ces conclusions, nous considérons que le modèle le plus adapté est le modèle étendu sans participation des ménages à la prise de décision du déménagement. Nous considérons aussi que le meilleur algorithme de classification pour la formation des groupes

sociaux et des types d'habitat est l'algorithme basé sur les nuages de particules. Autrement dit, le type de simulation le plus adapté à la reproduction de la distribution spatiale des groupes sociaux dans la ville de Bogotá est le type ME-DM-PART.

Dans les sections suivantes nous faisons une analyse de la sensibilité aux paramètres de ce type de simulation. D'autre part, à manière d'illustration des résultats graphiques des simulations, nous présentons dans l'Annexe V les diagrammes historiques et les cartes de validation des simulations du type ME-DM-PART avec le paramétrage de référence pour l'arrondissement « La Candelaria ».

### **11.3.2. Détermination du paramétrage**

Dans cette partie nous analysons la variabilité des résultats des simulations par rapport au paramétrage pour essayer de trouver les meilleures valeurs des paramètres et répondre ainsi au troisième objectif posé dans la section 11.1.

La stratégie utilisée est d'analyser la sensibilité du coefficient de corrélation par rapport aux variations des valeurs des différents paramètres de mobilité, de formation des groupes et de constitution des règles d'évolution. Nous faisons varier un paramètre à la fois et comparons les résultats aux résultats des simulations qui utilisent le paramétrage de référence (cf. section 11.2.3.4.).

Nous utilisons le type de simulation ME-DM-PART qui a été choisi dans la section précédente parmi les 9 types de simulation proposés initialement.

#### 11.3.2.1. Paramètres de mobilité

##### 11.3.2.1.1. Analyse de la sensibilité des résultats par rapport à la stratégie de déménagement

Dans les simulations avec le paramétrage de référence, les coûts de déménager ont été définis selon le cas de figure 1 (cf. section 11.2.3.1.1.), c'est-à-dire, en privilégiant l'hypothèse que les ménages préfèrent rester dans leur secteur urbain courant.

Dans cette partie, nous nous intéressons à l'effet d'autres cas de figure dans la définition des coûts de déménager.

Ici, nous considérons les deux autres cas de figure présentés dans la section 11.2.3.1.1. Nous avons effectué des simulations avec le modèle étendu ME-DM-PART. Chaque type de simulation a été exécuté 30 fois.

Les résultats des coefficients de corrélation pour ces simulations sont présentés dans le tableau et figure ci-dessous.

	La Candelaria		Santa Fé	
Coût de déménager	ZNCC	Ecart type du ZNCC	ZNCC	Ecart type du ZNCC
Cas 1 (préférence pour déménager dans le secteur courant)	0,812	0,014	0,920	0,010
Cas 2 (préférence pour les secteurs différents au secteur courant)	0,552	0,026	0,683	0,019
Cas 3 (coûts de déménagement aléatoires)	0,694	0,032	0,777	0,017

Tab 18. Coefficients de corrélation et écart type des simulations avec le type de simulation ME-DM-PART avec trois hypothèses différentes quant au coût de déménager (cf. section 11.2.3.1.1.). Chaque type de simulation a été exécuté 30 fois sur les données des arrondissements « La Candelaria » et « Santa Fé ».

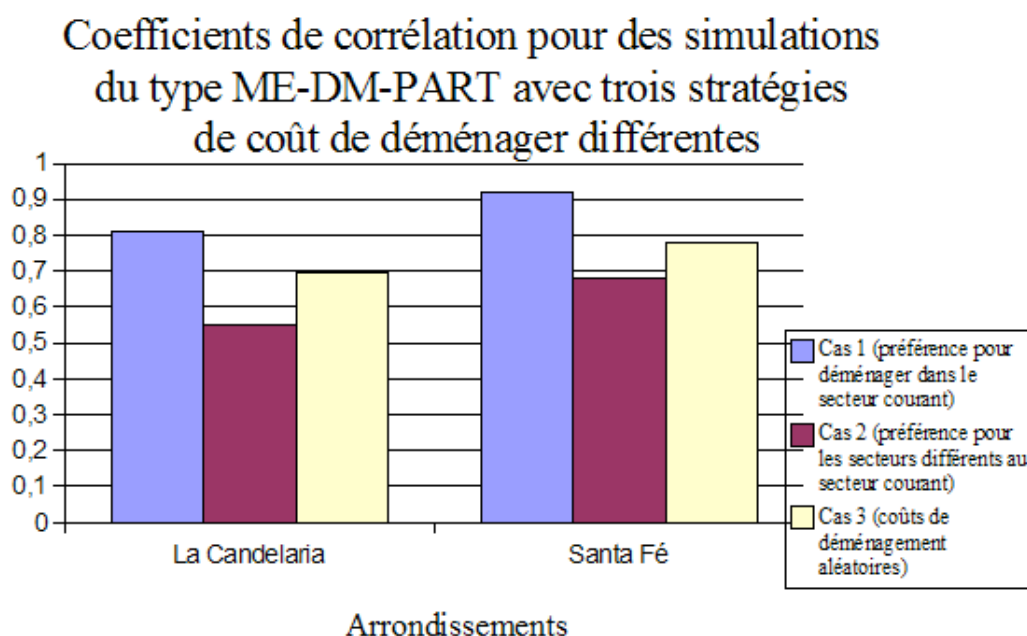


Figure 65. Graphique comparatif des coefficients de corrélation pour des simulations avec 3 hypothèses différentes pour la définition du coût de déménager. A gauche les résultats pour l'arrondissement « La Candelaria », à droite pour l'arrondissement « Santa Fé ».

Dans le graphique ci-dessus, nous présentons les coefficients de corrélation obtenus pour des simulations avec les trois stratégies de mobilité considérées dans la section 11.2.3.1.1. On observe dans ces graphiques une baisse significative des coefficients de corrélation par rapport à la 1<sup>ère</sup> stratégie. D'une part, cette baisse confirme la dépendance des résultats des simulations à la définition des coûts de déménager. D'autre part, on confirme l'hypothèse de la préférence des ménages pour déménager dans leurs secteurs urbains courants. L'hypothèse de la préférence pour déménager vers des secteurs différents du secteur courant obtient les plus mauvaises notes. Le cas des coûts de déménagement aléatoires est mieux noté que le cas

2. Dans le cas 3, le tirage aléatoire que nous avons utilisé pour générer les valeurs des coûts des déménagements entre les secteurs urbains est basé sur une distribution de probabilité normale centrée autour de la valeur 0,5. On obtient ainsi en moyenne une valeur proche de 0,5 pour les coûts, ce qui correspondrait à l'hypothèse que les ménages considèrent de manière équivalente toutes les secteurs urbains dans le cas du déménagement.

#### 11.3.2.1.2. Analyse de sensibilité par rapport à l'évolution de l'urgence pour résoudre la demande de groupes-sociaux-non-localisés

Dans les simulations avec le paramétrage de référence, on considère une augmentation du facteur de prix pour les groupes-sociaux-non-localisés de 1% entre recherches successives d'enchères ( $k = 1,01$  dans l'équation du *facteurPrix*, cf. section 6.2.2.2.2.). Pour étudier l'effet des variations dans les résultats par rapport aux variations dans le facteur d'augmentation, nous avons réalisé deux tests : le premier considère une diminution rapide du facteur de prix, c'est-à-dire une diminution de 50% du facteur entre recherches d'enchère successives ( $k = 0,5$  dans l'équation du *facteurPrix*, cf. section 6.2.2.2.2.). Dans le deuxième, on considère une augmentation rapide du facteur de prix, c'est-à-dire une augmentation de 100% du facteur entre deux recherches d'enchère ( $k = 2,0$  dans l'équation du *facteurPrix*, cf. section 6.2.2.2.2.). Nous avons effectué 30 exécutions avec chacun des deux nouveaux paramétrages. Les résultats sont présentés ci-dessous. Dans ces résultats on observe une diminution sensible des valeurs des coefficients de corrélation pour le cas des réductions successives de 50% du *facteurPrix* par rapport au paramétrage de référence (augmentations successives de 1 % du *facteurPrix*). Par contre, la différence entre les corrélations des simulations avec le paramétrage de référence et les corrélations des simulations avec des augmentations successives de 100% du *facteurPrix*, ne sont pas significatives. On peut conclure, à partir de ces résultats, que la valeur de  $k$  dans l'équation du *facteurPrix* (cf. section 6.2.2.2.2.) doit être supérieure à 1, c'est-à-dire que le *facteurPrix* doit augmenter sur des recherches successives d'enchères. Autrement dit, l'urgence pour résoudre la demande doit toujours augmenter. A partir des résultats obtenus, on ne peut pas fixer une valeur exacte pour le *facteurPrix*, mais on observe que pour le cas de référence, les résultats sont de bonne qualité.

	La Candelaria		Santa Fé	
	ZNCC	Ecart type du ZNCC	ZNCC	Ecart type du ZNCC
Augmentations successives de 1% du <i>facteurPrix</i>	0,812	0,014	0,920	0,010

Diminutions successives de 50% du <i>facteurPrix</i>	0,737	0,014	0,865	0,016
Augmentations successives de 100% du <i>facteurPrix</i>	0,837	0,023	0,931	0,017

Tab 19. Coefficients de corrélation et écart type des simulations avec le type de simulation ME-DM-PART avec trois valeurs différentes du *facteurPrix* pour les groupes sociaux non localisés (cf. section 6.2.2.2.2.). Chaque type de simulation a été exécuté 30 fois sur les données des arrondissements « La Candelaria » et « Santa Fé ».

### Coefficients de corrélation pour des simulations du type ME-DM-PART avec trois stratégies différentes d'évolution du *facteurPrix*

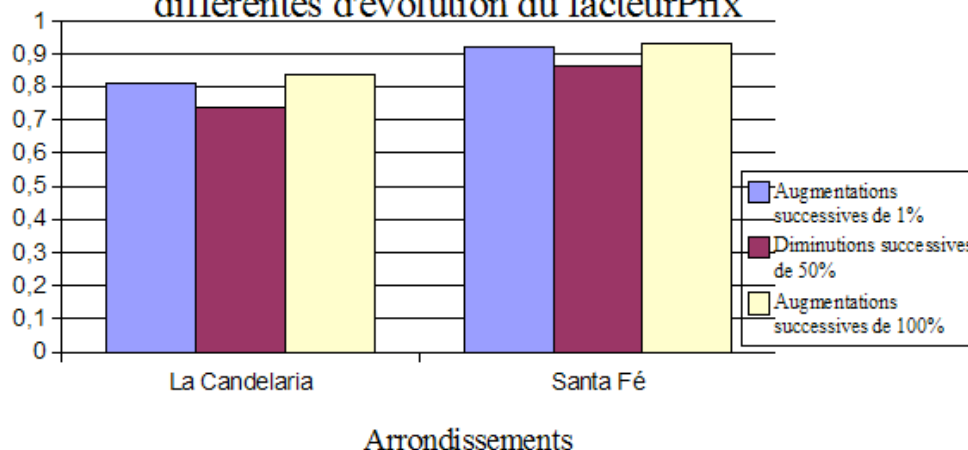


Figure 66. Graphique comparatif des coefficients de corrélation pour des simulations avec différents types d'évolution du *facteurPrix* (cf. section 6.2.2.2.2.). A gauche les résultats pour l'arrondissement « La Candelaria » à droite pour l'arrondissement « Santa Fé »

#### 11.3.2.2. Paramètres de formation de groupes

Dans le tableau ci-dessous, nous présentons les résultats du coefficient de corrélation pour des simulations avec les données de l'arrondissement « La Candelaria ». Dans la première ligne du tableau, nous présentons la valeur de corrélation pour le paramétrage de référence. Dans la deuxième ligne, nous présentons la valeur moyenne et l'écart type de la corrélation pour des simulations, où l'on a forcé l'algorithme de classification PART à trouver un nombre important de classes : pour les ménages on a fait varier le nombre de classes dans l'intervalle :  $[minK=70; maxK=100]$ , pour les logements on a fait varier le nombre de classes dans l'intervalle :  $[minK=20; maxK=50]$ . Dans les résultats présentés, on observe une diminution très significative de la valeur du coefficient de corrélation pour les simulations avec un



nombre important des classes. Malheureusement, on détient trop peu de résultats pour pouvoir établir une tendance globale, mais il semblerait que plus on s'approche du niveau microscopique dans la définition des classes, plus les erreurs des simulations sont grandes. Ceci s'explique par le fait que l'on a considéré des tendances globales comme moteur du mécanisme de mobilité. Ces tendances globales s'adaptent mieux à des niveaux intermédiaires suffisamment éloignés du niveau microscopique. Si l'on s'approche du niveau microscopique, il faut tenir compte des règles individuelles (comme on fait classiquement en micro-simulation) pour pouvoir représenter la mobilité des ménages.

	La Candelaria	
	ZNCC	Ecart type du ZNCC
Nombre de classes pour le paramétrage de référence	0,812	0,014
Nombre important de classes	0,604	0,018

Tab 20. Coefficients de corrélation et écart type du coefficient pour des simulations du type ME-DM-PART, en faisant varier le nombre de classes trouvés par l'algorithme de classification dans la formation des groupes. Chaque type de simulation a été exécuté 30 fois sur les données de l'arrondissement « La Candelaria ».

### 11.3.2.3. Paramètres de construction des règles d'évolution

Pour étudier la sensibilité des résultats au niveau d'abstraction utilisé dans la construction des règles d'évolution pour le type de simulation ME-DM-PART, nous avons construit de nouveaux ensembles de règles d'évolution avec un nombre de règles réduit par rapport au paramétrage de référence. A cet effet, nous avons utilisé l'algorithme SOM dans la classification nécessaire pour la construction règles. Le paramétrage de l'algorithme SOM utilisé a été présenté dans la section 11.2.3.3.2

Nous avons exécuté des simulations du type ME-DM-PART avec les nouvelles conditions et nous avons obtenu les résultats présentés dans la deuxième ligne du tableau et dans le graphique ci-dessous. On observe une diminution très grande du coefficient de corrélation pour les simulations avec le nouvel ensemble de règles. On peut conclure que la définition des règles d'évolution a une importance capitale, les règles d'évolution doivent se rapprocher le plus possible du niveau microscopique pour obtenir de bons résultats de simulation.

	La Candelaria		Santa Fé	
	ZNCC	Ecart type du ZNCC	ZNCC	Ecart type du ZNCC
Règles d'évolution avec le Clustering Basique	0,812	0,014	0,920	0,010
Règles	0,525	0,201	0,483	0,050

d'évolution avec l'algorithme SOM				
-----------------------------------	--	--	--	--

Tab 21. Coefficients de corrélation et écart type des simulations avec le type de simulation ME-DM-PART avec pour deux ensembles différents de règles d'évolution (cf. section 11.2.3.3.). Chaque type de simulation a été exécuté 30 fois sur les données des arrondissements « La Candelaria » et « Santa Fé ».

### Coefficients de corrélation pour des simulations du type ME-DM-PART avec deux ensembles différents de règles d'évolution

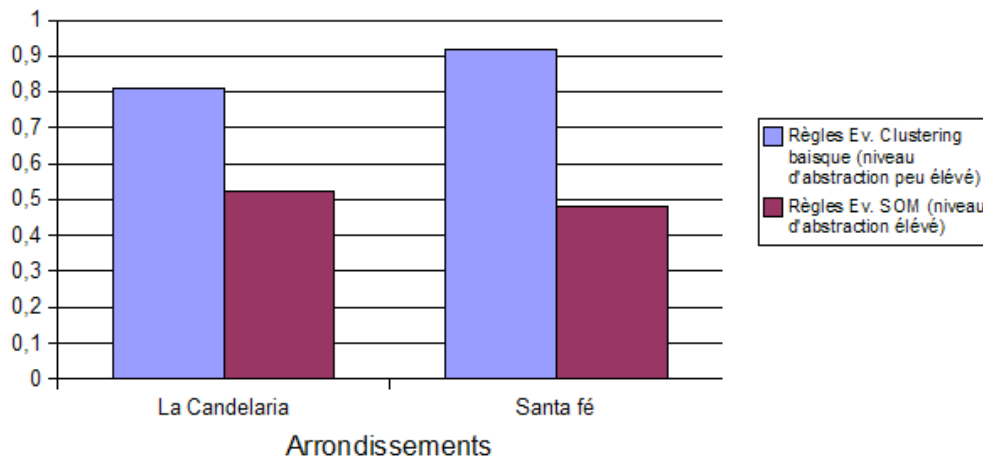


Figure 67. Graphique comparatif des coefficients de corrélation pour des simulations du type ME-DM-PART avec deux ensembles différents des règles d'évolution construits à des niveaux d'abstraction différents (cf. section 11.2.3.3.). A gauche les résultats pour l'arrondissement « La Candelaria » à droite pour l'arrondissement « Santa Fé »

### 11.3.3. Résultats graphiques

Dans l'Annexe V, nous présentons à titre d'exemple les résultats graphiques des simulations réalisées avec le type de simulation ME-DM-PART et avec le paramétrage de référence pour l'arrondissement « La Candelaria ». Nous y présentons principalement les diagrammes historiques des groupes sociaux et des types d'habitat (cf. Annexe A.5.1), la caractérisation automatique basée sur le calcul des valeurs-test des groupes sociaux et des types d'habitat (cf. Annexe A.5.2) et les cartes de validation de la distribution spatiale des groupes sociaux et de matrices d'occupation des types d'habitat (cf. Annexe A.5.3).

## 11.4 Discussion sur les résultats des simulations

A partir des résultats obtenus, nous avons principalement démontré que le fait de considérer des groupes sociaux et des types d'habitat comme entités principales dans la modélisation des dynamiques de mobilité résidentielles permet de reproduire la distribution spatiale des groupes sociaux au niveau de la ville avec une très bonne précision. On a trouvé aussi que les principaux aspects à considérer dans ce type de modélisation sont :

- Il faut tenir compte d'un niveau d'abstraction élevé dans la constitution des groupes sociaux et des types d'habitat. En effet, nous avons montré que plus le nombre de groupes sociaux et de types d'habitat considérés est petit (et en conséquence plus le niveau d'abstraction est élevé), meilleurs sont les résultats des simulations.
- Il faut tenir compte du niveau microscopique pour pouvoir représenter l'évolution de la population de ménages et du parc de logement. Il est nécessaire de définir des règles d'évolution très proches du niveau microscopique. Nous avons trouvé que lorsqu'on augmente le niveau d'abstraction de ces règles (on considère un nombre petit de règles) la qualité des simulations est diminuée.
- Il n'est pas nécessaire de considérer les ménages dans la prise de décision de déménager. Ce processus est assez bien représenté à partir des interactions entre groupes sociaux et types d'habitat.
- La dépendance démontrée des résultats des simulations aux paramètres de mobilité valide la prise en compte des coûts de déménager et d'un facteur d'évolution du prix dans le mécanisme de mobilité. La correcte définition de ces facteurs est très importante et a un effet considérable dans les résultats de la simulation.

D'autre part, nous avons montré que la stabilité du mécanisme de mobilité proposé est très bonne et les résultats quant à la précision des simulations sont très encourageants. La qualité des résultats vérifie les deux hypothèses fondamentales posées dans la définition du mécanisme de mobilité, c'est-à-dire, que l'on peut exprimer la relation entre groupes sociaux et types d'habitat par un système de préférences localisé et que la mobilité résidentielle dépend de ces préférences mais aussi de l'offre et de la demande de logements libres. Ces hypothèses reposent sur les résultats obtenus dans les travaux sur Bogotá de [Piron et al. 2005 ; 2007] et qui se situent dans une approche d'écologie urbaine (Park, Burgess & Mackensie, 1925 ; Burgess & Park 1926 ; Park 1929 ; Wirth 1938 ). Dans le cas spécifique de Bogotá, nous avons trouvé que la stratégie résidentielle qui s'adapte le mieux à la situation

des deux arrondissements dans la période 1973 – 1993 est la préférence pour déménager dans le secteur courant du ménage. Cette hypothèse a aussi été confirmée par d'autres travaux sur Bogotá , notamment [Piron et al. 2007], où l'on a trouvé qu'à Bogotá, les réseaux familiaux sont souvent de proximité et sont déterminants dans le choix de localisation.

D'un point de vue plus technique, nous avons trouvé que la précision des simulations est très liée à la qualité de l'algorithme de classification utilisé dans la formation de groupes. Quant à la qualité de la classification, nous avons observé que le meilleur algorithme<sup>23</sup> parmi les algorithmes utilisés est l'algorithme SOM. Toutefois, une grande limitante de cet algorithme est la dépendance à l'ordre de présentation des exemples, de même que la configuration de la grille et les paramètres initiaux (*initTau* par exemple). Cette dépendance rend difficile la reproductibilité des classes sur un même jeu de données. D'autre part, l'algorithme basé sur les nuages de particules montre aussi une très bonne performance quant à la qualité de la classification et quant à la précision des résultats de simulations. Cet algorithme a l'avantage d'être un algorithme beaucoup plus stable : pour un même jeu de données et pour différentes exécutions de l'algorithme la probabilité de retrouver toujours la même partition est très grande. La principale limitante de l'algorithme basé sur les nuages de particules est le temps d'exécution. A titre d'exemple, dans le tableau ci-dessous, nous présentons le temps d'exécution nécessaire pour précalculer les classes utilisées dans les simulations avec le modèle étendu pour les données de « La Candelaria ». La détermination des classes avec les différents algorithmes a été réalisée sur un ordinateur iMAC G5<sup>24</sup> avec un processeur de 2,0 GHz et 512 Mio de mémoire RAM. On observe clairement que pour ce cas particulier la détermination des classes avec l'algorithme SOM est environ 59 fois plus rapide que pour l'algorithme basé sur les particules.

	Temps d'exécution (minutes)
SOM	11
Multi-Kmeans	166
Particules	643

Tab 22. Temps d'exécution pour précalculer les classes sur les données de « La Candelaria » et utilisés dans les simulations avec le modèle étendu. Les exécutions ont été réalisées sur un iMAC G5 avec un processeur de 2,0 GHz et 512 Mio de mémoire RAM.

Une analyse comparative des différents algorithmes de classification automatique utilisés s'avère fondamentale. Ce type de validation doit faire appel à des analyses de stabilité par rééchantillonnage comme le *bootstrap* [Efron & Tibshirani, 1994]. Une première analyse de la

<sup>23</sup> A condition de réduire la variabilité dans les partitions liée à l'algorithme de classification

<sup>24</sup> cf. <http://www.apple.com/fr/imac/>

performance de l'algorithme de classification basé sur les nuages de particules a été faite dans [Gil-Quijano & Piron, 2007]. Toutefois, une analyse plus approfondie des performances des algorithmes de classification utilisés est hors du sujet traité dans ce document. Une telle analyse fait partie de nos travaux futurs. De même, il semble intéressant d'approfondir la piste des algorithmes de classification déterministes pour améliorer la reproductibilité du mécanisme de formation de groupes, ainsi que de rendre l'algorithme basé sur les nuages de particules plus performant en terme de temps de calcul.

## **Partie IV Discussion**

Les mécanismes présentés dans ce document permettent de reconstruire, à partir de deux années dont la distribution de la population spatiale est connue, l'évolution de cette distribution entre les deux années. Ces mécanismes permettent d'avoir une vision synthétique de la ville par l'utilisation de groupes sociaux et de types d'habitat comme entités principales de modélisation. En effet, d'une part, dans le cadre de l'écologie urbaine (Vicari, 1981 ; Grafmeyer & Joseph, 1990 ; Roncayolo 1990 ; Grafmeyer 1994 ), nous avons implémenté un mécanisme de redistribution de la population d'une ville, basé sur la concurrence entre groupes sociaux pour l'acquisition de l'espace (acquisition de groupes de logements libres). Et d'autre part, nous avons représenté l'évolution de la population et du parc de logement, à l'aide de règles générales qui s'opposent à la vision classique utilisée en micro-simulation basée sur la définition de règles individuelles d'évolution.

La vision synthétique adoptée dans notre travail, permet de représenter de manière simple les interactions entre entités. En effet, lorsqu'on se situe au niveau des groupes, les interactions entre entités répondent à des logiques globales moins nombreuses et plus faciles à identifier et à modéliser que les interactions au niveau microscopique. D'autre part, la quantité de données nécessaire pour définir des règles générales d'évolution est beaucoup plus réduite en comparaison avec le cas de règles d'évolution/transition microscopiques. Dans le cas de règles microscopiques, il est souvent nécessaire d'avoir recours à des enquêtes additionnelles, qui permettent de définir des fonctions stochastiques de transition d'état basées sur les profils individuels. Dans le mécanisme présenté ici, nous construisons les règles d'évolution, uniquement à partir des distributions de population initiale et finale pour la période d'étude.

La concurrence entre groupes sociaux pour l'espace et les règles générales d'évolution constituent le moteur de l'évolution de la distribution spatiale de la population de la ville. Grâce à ce double mécanisme, la distribution de la population de la ville évolue de manière auto-organisée sans avoir recours à la définition d'une finalité politique ou économique explicite qui devrait être décrite sous forme d'une fonction à optimiser ou bien d'un équilibre à atteindre, comme il est par exemple le cas dans le cadre de la modélisation des mobilités résidentielles par la théorie de jeux (Piron et *al.* 2003). D'autre part, par rapport aux travaux précédents en modélisation des mobilités résidentielles à Bogotá (Piron et *al.* 2004, 2006 ; Piron 2005), nous avons pu considérer les groupes sociaux et les types d'habitat d'une manière dynamique. En effet, contrairement aux travaux précédents, nous avons considéré des groupes sociaux et des types d'habitat dont la structure, la taille et la description (caractéristiques du profil moyen) évoluent dans le temps.

Du point de vue de la simulation multi-agent, la considération d'un niveau intermédiaire formé par des structures telles que les groupes sociaux et les types d'habitat introduit un certain nombre de problèmes qui relèvent de la nature abstraite des structures introduites. Ces problèmes qui, à notre connaissance, n'ont pas été traités dans la littérature, portent principalement sur la formation et l'évolution automatiques de ces structures abstraites ainsi que sur la définition des interactions entre ces structures et les micro-entités qui les composent. Nous avons souligné ces problèmes dans l'introduction de ce document, où nous considérons ces structures comme des agents composés abstraits. Les mécanismes présentés dans ce document donnent une réponse à chacun des ces problèmes.

Premièrement, grâce à l'utilisation des méthodes de classification automatique, nous avons proposé un mécanisme de formation et d'agentification des agents composés abstraits. Les agents composés abstraits (groupes sociaux et types d'habitat) considérés ici, sont formés à partir des classifications automatiques des micro-entités (ménages et logements). Ces classifications considèrent les caractéristiques socio-démographiques des micro-entités pour former des groupes de micro-entités semblables. L'agentification consiste dans ce cas en la création d'un agent composé pour chaque groupe de micro-entités trouvé dans les classifications. La méthode de formation et d'agentification d'agents composés abstraits basée sur la classification des profils des micro-entités est très facilement généralisable à d'autres types de problèmes qui considèrent des micro-entités décrites par un nombre important de variables. Les choix des algorithmes de classification, de la représentation des caractéristiques (par exemple sous forme disjonctive complète) et des critères de similarité entre données (par exemple distance entre profils) restent toutefois une responsabilité du modélisateur. Comme nous avons démontré grâce aux résultats des simulations, dans le choix de l'algorithme de classification, on ne doit pas seulement considérer des critères « locaux » tels que la qualité des classifications mais aussi des résultats de performance globale comme par exemple la précision obtenue dans les résultats des simulations. Les simulations, dans ce cas, sont utilisées en premier lieu pour « calibrer » ou « choisir » le meilleur algorithme de classification et son paramétrage dans la formation et l'agentification des agents composés abstraits.

Deuxièmement, quant à la question de l'évolution des agents composés abstraits, à partir des résultats expérimentaux, nous avons trouvé que dans le cas de groupes sociaux et de types d'habitat, il est nécessaire de considérer d'abord l'évolution des populations de micro-entités (ménages et logements) pour ensuite transmettre les effets de ces évolutions aux agents composés. L'évolution des populations de micro-entités (évolutions microscopiques) sont la



conséquence de l'exécution des règles globales d'évolution. La transmission des effets des évolutions microscopiques au niveau des agents composés est faite ici par un double mécanisme, qui consiste à la re-classification des populations des micro-entités et la mise en relation des nouvelles classes trouvées avec les anciennes (dans la dernière année de simulation). De cette manière, d'une part, la structure (micro-entités qui les composent) et les caractéristiques moyennes des agents composés évoluent sur plusieurs années successives de simulation. D'autre part, la disparition/apparition d'anciens/nouveaux agents composés a été traitée de manière « indirecte » lors de la mise en relation des différents ensembles de classes trouvées entre plusieurs années successives de simulation. Les différences en nombre entre deux ensembles de classes signifient dans notre système l'apparition ou la disparition de nouveaux agents composés sur des années successives de simulation. Cette première approche doit être approfondie, et doit considérer dans un deuxième temps, des conditions plus liées à l'évolution même du profil moyen du groupe et considérer par exemple des situations de « rupture » où l'on considère la disparition du groupe ancien et la création d'un nouveau lors, par exemple, du changement au-delà d'un seuil du profil moyen du groupe.

D'autre part, nous considérons que la définition des interactions entre agents composés abstraits, ainsi que des interactions multi-niveaux (entre agents composés et micro-entités) dépendent fortement du problème modélisé. Ici, il s'agissait d'une part, dans le cadre de l'écologie urbaine, de représenter la concurrence entre groupes sociaux pour l'acquisition de l'espace urbain. Le mécanisme basé sur des enchères de logement proposé s'adapte naturellement pour représenter une telle concurrence. D'autre part, quant aux interactions entre agents composés et micro-entités, nous avons considéré ici principalement des interactions descendantes, par lesquelles les groupes sociaux relocalisent les ménages qui les forment. Nous avons testé l'introduction d'interactions ascendantes, grâce auxquelles les ménages auraient un certain poids dans la prise de décision du déménagement. Mais les expériences réalisées ont démontré que les résultats ne changent pas de manière significative lors de la prise en compte des interactions ascendantes. Toutefois, les interactions ascendantes considérées ici sont très simplifiées et d'une certaine manière trop liées à l'appartenance au groupe social. Il est possible que des interactions « plus individualisées » basées sur des règles individuelles (telles que les règles utilisées en micro-simulation) aient un impact important sur la qualité des résultats des simulations. Cependant, la détermination de telles règles nécessite un nombre très important de données et leur introduction augmenterait la complexité du point de vue informatique mais aussi du point de vue de l'analyse des résultats des simulations.

*La prise en compte de l'espace dans la modélisation*

Comme nous l'avons vu dans la partie I, lorsqu'on s'intéresse à la constitution d'agents composés « physiques » (structures hydrologiques, robots, organismes pluricellulaires, etc.) les mécanismes de constitution se basent sur les interactions locales des micro-agents localisés dans un espace physique. Lorsqu'on s'intéresse à la constitution d'agents composés abstraits (groupes sociaux, types d'habitat), l'espace « physique » (s'il en existe un) doit être pris en compte avec précaution. Dans le cas particulier de la formation de groupes sociaux et de types d'habitat, l'espace physique est l'espace urbain, où les logements sont localisés et où les ménages habitent. Si l'on considère des interactions locales sur l'espace urbain pour constituer les groupes sociaux ou les types d'habitat, la composante spatiale prend une place trop importante par rapport aux caractéristiques des micro-agents. En effet, comme les groupes sociaux et les types d'habitat sont ici des typologies, l'espace ne peut pas être imposé *a priori*. Ce qui nous intéresse est de trouver des groupes de ménages ou de logements semblables quant à leurs caractéristiques socio-démographiques. La localisation spatiale peut faire partie de ces variables, mais son importance ne doit être découverte qu'*a posteriori*. Une approche multi-agent « classique » basée sur des interactions localisées dans l'espace urbain, s'avère inadéquate pour constituer des typologies, en raison de la forte importance donnée à la localisation des agents dans l'espace. Par contre, une approche par la classification automatique permet de rendre indépendante la constitution des typologies de la localisation spatiale de micro-agents. La localisation spatiale peut être considérée dans la description des micro-agents et son importance est éventuellement relevée *a posteriori* par la classification.

D'autre part, lorsqu'on utilise les systèmes multi-agents dans la classification automatique, il est nécessaire d'introduire un nouvel espace appelé *espace d'interaction* qui est différent de l'espace physique (espace urbain dans notre cas). L'espace d'interaction définit le support de la communication et de l'interaction entre les agents classificateurs et les données (dans le cas des algorithmes basés sur les colonies de fourmis) ou entre les agents qui représentent les données (dans le cas des algorithmes basés sur les nuages des particules). C'est dans cet espace que les données sont déplacées (ou se déplacent) afin de les regrouper dans des amas de données similaires qui constituent par la suite les classes. Le fait de ne considérer que des interactions locales dans l'espace d'interaction a aussi des effets pervers, notamment dans le cas des algorithmes basés sur les nuages de particules. En effet, les résultats des classifications par ces méthodes dépendent de la définition de la forme et de la taille du voisinage sur l'espace d'interaction. En général, on observe une augmentation du nombre de classes avec la réduction de la taille du voisinage. Le fait de considérer des interactions locales rend aussi les classifications très dépendantes des localisations initiales (généralement

aléatoires) des données dans l'espace d'interaction. Malheureusement, il n'existe pas de stratégie valide et simple pour définir la forme et la taille de ce voisinage. Dans le cas de l'algorithme de classification par les nuages de particules que nous avons développé (cf. section 8.4.), nous avons considéré que tous les agents-données interagissent avec tous les autres, ce qui revient à considérer un voisinage infini (ou du moins suffisamment grand pour inclure tous les agents-données). La solution que nous avons adoptée permet de supprimer la difficulté pour définir la forme du voisinage et rend plus indépendante la classification aux localisations initiales des données dans l'espace de classification. L'effet négatif de ce type de voisinage infini est que le nombre d'interactions est très grand, ce qui augmente la complexité algorithmique<sup>25</sup> et en conséquence le temps d'exécution de l'algorithme.

#### *La prise en compte du temps dans la modélisation*

Par rapport à la gestion du temps, il existe deux types de simulations : les simulations dirigées par une horloge et les simulations dirigées par des événements. Dans le premier cas, le temps est discret, il est divisé en intervalles dont la longueur est appelée « pas de temps ». A chaque pas de temps, toutes les variables du système sont actualisées en utilisant les mêmes fonctions de transitions à chaque fois [Gilbert & Troitzch, 1999]. Dans les simulations dirigées par des événements, l'élément central n'est pas la définition d'un pas de temps mais la prise en compte d'événements, c'est-à-dire, d'occurrences instantanées qui altèrent l'état du système (cf. *queuing models* dans Gilbert & Troitzch, 1999). Dans le cas de simulations dirigées par une horloge, le problème fondamental est de définir le pas de temps. En effet, pour des pas de temps trop longs, des événements importants de durée courte peuvent ne pas être pris en compte, tandis que pour des pas de temps trop petits, on peut augmenter de manière considérable le temps d'exécution de la simulation. Un autre problème dans les simulations dirigées par une horloge est l'exécution d'événements simultanés (exécution synchrone) qui induit des problèmes de coordination et de partage de ressources. Dans le cas de simulations dirigées par des événements, le problème de définition du pas de temps ne se pose plus : la simulation évolue par l'exécution successive d'événements qui sont stockés dans un « agenda ». Les événements passés sont supprimés de l'agenda et leur exécution peut produire l'ajout de nouveaux événements à l'agenda. Les événements sont ainsi exécutés de manière asynchrone, c'est-à-dire, un événement à la fois, ce qui évite le problème d'exécution simultanée de plusieurs événements. Le problème évident de l'exécution asynchrone des

---

<sup>25</sup> La complexité des algorithmes basés sur les nuages de particules est de  $O(N \times M)$ , où  $N$  est le nombre de données et  $M$  le nombre (variable) de données dans le voisinage. Si l'on considère un voisinage infini, la complexité algorithmique est de  $O(N)$ .

événements est le biais dû à l'ordre d'exécution des événements. En effet, si on utilise toujours le même ordre d'exécution des événements, on restreint les possibles évolutions du système. Pour palier ce problème, les événements doivent être exécutés suivant un ordre aléatoire.

Dans notre système, nous avons utilisé une double approche pour effectuer les simulations : d'une part, nous avons utilisé un temps discret pour exécuter les règles d'évolution et d'autre part, la concurrence entre groupes sociaux pour l'accès au logement a été implémentée de manière asynchrone. L'utilisation d'un temps discrétisé pour exécuter les règles d'évolution produit une évolution graduelle et uniforme de la population des ménages et du parc du logement tout au long de la période de simulation. Cette manière de traiter l'exécution des règles d'évolution était suffisante dans un premier temps pour effectuer la calibration du modèle. Toutefois, une utilisation future à des fins prédictives, doit aussi tenir compte des événements ponctuels dans le temps, comme par exemple, la construction rapide d'un ensemble immobilier. Un traitement hybride du temps (temps discret plus des événements ponctuels) quant à l'exécution des règles d'évolution doit être alors implémenté. De plus, le pas de temps d'un an utilisé par défaut dans l'exécution des règles d'évolution a été choisi à partir du compromis entre le nombre d'événements considérés et le temps d'exécution des simulations. Néanmoins, une analyse plus approfondie de la taille adéquate du pas de temps doit être effectuée. Cette analyse peut être réalisée en suivant le même schéma utilisé dans la partie III pour tester la dépendance des résultats des simulations aux différents paramètres, c'est-à-dire, en choisissant le pas de temps (parmi plusieurs possibilités) qui donne les meilleurs résultats de simulation par rapport à une distribution de population réelle.

#### *Travail futur*

Dans l'état courant, notre modèle a été calibré pour reproduire la distribution de la population de Bogotá à la fin de la période de simulation par rapport à une distribution réelle. Notre objectif futur est de pouvoir utiliser ce modèle à but prédictif. Pour cet objectif, nous étudions actuellement, l'introduction de règles d'évolution définies par l'utilisateur et non pas générées de manière automatique. L'objectif de l'introduction des règles définies par l'utilisateur est de permettre la définition des scénarii ponctuels tels que la construction d'un ensemble immobilier d'un certain type dans un secteur urbain spécifique ou l'arrivée dans la ville de populations hypothétiques de ménages. Ces types de règles présentés dans [Gil Quijano & al., 2007b.] se basent sur la définition de quotas d'ajout/suppression de logements ou de ménages dont les profils sont construits par l'utilisateur du système à l'aide d'opérateurs logiques. Par

l'introduction de ces types de règles, l'utilisateur peut étudier deux types possibles de scénario:

- étudier l'effet sur la distribution spatiale de la population de l'action d'ajouter/supprimer des logements de certains types dans certains secteurs urbains.
- étudier l'effet sur la distribution spatiale de la population de l'action d'ajouter/supprimer des ménages appartenant à certains profils.

Malheureusement, la capacité « prédictive » du modèle n'a pas pu être testée pour l'instant en raison du manque de données. En effet, nous ne disposons que de deux recensements (1973 et 1993)<sup>26</sup> de population valides à Bogotá, qui nous ont permis de calibrer le modèle. Pour pouvoir tester la capacité prédictive du modèle, il nous faudrait un troisième recensement, pour pouvoir par exemple tester la validité des règles d'évolution générées de manière automatique à partir des données de deux recensements pour reproduire la population d'un troisième recensement, ou de tester différents scénarii à partir des règles d'évolution définies par l'utilisateur.

D'autre part, un problème fondamental dans notre modèle est la prise en compte d'un nombre très grand de données qui représentent les ménages et les logements. Le fait d'utiliser des groupes sociaux et des types d'habitats a réduit la complexité informatique quant à la quantité d'interactions entre entités. Mais la nécessité, démontrée par les expérimentations réalisées, de considérer le niveau microscopique dans la prise en compte de l'évolution des groupes sociaux et des types d'habitat, fait que les données représentant les ménages et les logements sont présentes tout au long de la simulation, ce qui ralentit l'exécution (notamment lors des classifications) et exige une capacité importante de ressources informatiques en termes de mémoire. En raison de ces limites, nous n'avons pu réaliser de simulations que sur certains arrondissements de la ville de Bogotá et non pas sur sa totalité<sup>27</sup>. Il s'avère très important d'approfondir d'une part l'optimisation des algorithmes (particulièrement les algorithmes de classification automatique), mais aussi de lancer de nouvelles pistes de recherche, par exemple par l'utilisation d'algorithmes de classification hiérarchique qui introduiraient des niveaux intermédiaires d'abstraction entre les niveaux mésoscopique et microscopique. D'autre part, il est aussi important d'approfondir la recherche de simplifications pertinentes

---

<sup>26</sup> En réalité il existe, deux autres recensements réalisés en 1985 et en 2005 à Bogotá. Mais d'une part le recensement de 1985 est inexploitable dans notre travail car la définition des secteurs de recensement est différente de celle utilisée dans les recensements de 1973 et 1993. D'autre part, les résultats du recensement de 2005 ne sont pas encore disponibles au public.

<sup>27</sup> Les simulations les plus grandes réalisées ont pris en compte environ 30000 ménages et un nombre égal de logements. La ville de Bogotá en 1993 comptait avec environ 1,5 millions de ménages et de logements.

par rapport à la quantité de données nécessaires dans une simulation : par exemple en travaillant sur des échantillons représentatifs de la population de ménages et du parc du logement, ou encore de réduire la taille en mémoire des données utilisées par l'utilisation des techniques de compression ou des nouvelles formes de représentation de données qui n'aient pas un effet important sur la performance des algorithmes de classification. D'autres mesures tendant à réduire le temps de calcul à d'autres niveaux sont possibles, par exemple au lieu de re-classifier les populations de micro-entités à chaque pas de temps, on peut conserver les classes pendant un certain nombre de pas de temps, en ajoutant/supprimant des micro-entités à ces classes. Les classes ne seraient recalculées que lorsque une mesure de la qualité globale de la classification (par exemple l'inertie intra-classes) atteindrait un seuil critique donné.

D'un autre côté, une limite observée de l'utilisation de la distance de Hamming (et en conséquence de la distance euclidienne qui dépend directement de la distance de Hamming, cf Annexe III) entre profils de micro-entités, dans la classification est qu'elle donne un poids plus important aux variables avec un nombre réduit de modalités. Néanmoins le calcul de la distance de Hamming est très performant. Une possibilité pour éliminer le biais introduit par la distance de Hamming est d'utiliser la distance de *Khi2* [Lebart et *al.*, 2006] qui permet de normaliser les distances par rapport au nombre de modalités de chaque variable et à la taille de chaque modalité. Le problème de l'introduction de la distance de *Khi2* est que son calcul est beaucoup plus complexe que celui de Hamming, difficilement optimisable et produit par conséquence des augmentations très importantes du temps d'exécution des classifications.

D'un point de vue plus technique, actuellement, les profils de ménages et de logements sont représentés dans l'implémentation par des variables de type *long*, ce qui dans le langage C++ correspond à *64 bits* de mémoire. On peut alors représenter un maximum de *64* modalités pour décrire les ménages ou les logements. Ce nombre a été largement suffisant pour représenter les données utilisées dans les simulations présentées dans la partie III. Toutefois, ce nombre de modalités peut être insuffisant pour représenter d'autres données (d'une autre ville ou si l'on décide d'intégrer des nouvelles variables dans la représentation des ménages ou des logements dans la ville de Bogotá). L'utilisation de cette représentation permet d'une part de faire une utilisation efficace de la mémoire et d'autre part de permettre un calcul performant de la distance de Hamming (à partir des opérations bit à bit, cf. Annexe A3.1.). Si l'on doit considérer un nombre de modalités plus grand que *64* pour représenter les ménages ou les logements, il faudrait changer le type de représentation, tout en respectant la performance dans le calcul de la distance et l'utilisation efficace de la mémoire. La

considération d'une nouvelle représentation des données est naturellement liée aux problèmes de la taille de données et du calcul performant et non biaisé de la distance, déjà exposés dans les paragraphes précédents.

Finalement, il serait aussi très intéressant d'adapter notre modèle à la représentation de la distribution spatiale dans d'autres villes. Tout en changeant les hypothèses par exemple quant à la définition des coûts des déménagements, on pourrait tester la validité de l'utilisation des groupes sociaux et des types d'habitat ainsi que de l'hypothèse de la concurrence pour l'espace (représentée par le mécanisme basé sur les enchères de logement) dans un contexte différent et éventuellement en tirer des règles généralisables à plusieurs « types » de villes.

# **Annexes**



## Annexe I. Description des caractéristiques des ménages et logements de la ville de Bogota

Dans les deux tableaux ci-dessous, nous présentons les caractéristiques et les modalités considérées dans la description des ménages (le ménage est décrit par les caractéristiques du chef de ménage) et des logements dans la ville de Bogotà. Les variables continues, notamment l'âge, ont été préalablement découpées en modalités.

Caractéristique	Modalité	Caractéristique	Modalité
Genre	Homme	Alphabétisation	Non alphabète
	Femme		Alphabète
Age	- 25 ans	Activité	Recherche emploi
	- 30 ans		Non actif
	- 35 ans		W semaine précéd
	- 40 ans	Statut d'activité	Non actif
	- 45 ans		Patron
	- 50 ans		Ouvrier employé
	- 60 ans		Indépendant
	+ 60 ans		Emploi domestique
Lieu de naissance	Non autochtone	Nombre de personnes	1pers/men
	Autochtone		2pers/men
Lieu de résidence 5 ans avant enquête	Lieu 5 ans Bogota		3pers/men
	Lieu 5 ans ailleurs		4pers/men
Etat civil	Célibataire		5pers/men
	Marié		6pers/men
	Union libre		+7pers/men
	Séparé/divorcé	Niveau Educatif	Sans niveau éduc.
	Veuf		Primaire
Statut d'occupation Du logement	Propriétaire		Secondaire
	Occupation lgt autre		Supérieur
	Locataire		

Tab. 23 Description des 11 caractéristiques et 43 modalités caractérisant les ménages. Source : recensements de 1973 et 1993 à Bogotà DANE

Caractéristique	Modalité
Matériel des murs	Mur brique
	Mur autre
Matériel du sol	Sol terre
	Sol ciment
	Sol bois
	Sol autre synth.
Présence de cuisine	Oui
	Non
Eau dans la cuisine	Eau robinet
	Eau autre
Nombre de pièces	1piec/men.
	2piec/men.
	3piec/men.
	4piec/men.
	+5piec/men.

Caractéristique	Modalité
Connexion eau	Oui
	Non
Connexion électricité	Oui
	Non
Connexion égout	Oui
	Non
Type sanitaire	Sanitaire classique
	Sanitaire autre
Utilisation sanitaires	Sanitaire partagé
	Sanitaire exclusif
Type du logement	Cuarto
	Maison
	Appartement
	Autre Logement

Tab. 24 Description des 11 caractéristiques et 29 modalités caractérisant les logements. Source : recensements de 1973 et 1993 à Bogotá DANE

## Annexe II. Représentation des profils

Les caractéristiques qui décrivent les ménages et les logements présentées dans l'annexe précédente sont découpées en modalités. De cette manière, on peut facilement les représenter sous forme disjonctive complète [Lebart et *al.*, 1997]. Dans la représentation sous forme disjonctive complète, comme nous l'avons présenté dans la section 3.1 (première partie du document), il s'agit de transformer la description de chaque micro-entité en vecteur binaire à  $p$  (où  $p$  est le nombre de modalités) composantes. Pour le cas de Bogotá  $p=43$  pour les ménages et  $p=29$  pour les logements. Dans ce vecteur, la valeur de chaque composante est déterminée par :  $v_d(j) = (t_{dj})$  où  $t_{dj} = 1$  si la donnée dispose de la modalité  $j$  ( $j=1, \dots, p$ ) et 0 sinon. Nous appelons ce vecteur binaire le profil de la micro-entité.

Dans notre cas, nous réalisons une transformation additionnelle : nous représentons chaque profil par un entier. Cet entier correspond à la conversion du vecteur binaire à la base décimale. Cette conversion est faite de manière classique comme il suit :

Entrée :  $v_d$  : la représentation sous forme disjonctive complète du profil

Sortie :  $ent$  : la représentation sous forme d'entier du profil

DEBUT

10.  $ent \leftarrow 0$

11. POUR chaque composante  $v_d(j)$

FAIRE

a.  $ent \leftarrow (ent * 2) + v_d(j)$

FIN\_POUR

12. renvoyer  $ent$

FIN

Encadré 25. Conversion d'un profil sous forme conjonctive complète en profil sous forme d'entier

Le fait de représenter chaque profil comme un entier, nous permet de calculer facilement les distances entre profils, notamment la distance de Hamming (voir annexe III). Dans l'implémentation, nous représentons chaque profil par un *long long*. En C++ ce type de variable dispose de 64 bits de mémoire, ce qui nous permet de représenter jusqu'à 64 modalités. Ce nombre est largement suffisant pour le cas de Bogotá, où l'on considère 43 modalités pour les ménages et 29 modalités pour les logements.

## Annexe III. Calculs de distances entre profils

Dans cette annexe, nous présentons deux mesures de distance utilisées dans les modèles proposées essentiellement dans le mécanisme de formation de groupes.

### A3.1. Distance de Hamming

La distance de Hamming [Hamming, 1950] est une mesure qui calcule le nombre de différences entre deux suites de symboles de la même longueur. Pour pouvoir calculer la distance de Hamming, il est nécessaire de connaître l'ensemble des valeurs que peuvent prendre les symboles. Cet ensemble est appelé alphabet, il s'agit d'un ensemble fini. Dans le cas binaire l'alphabet  $A$  ne contient que deux valeurs :  $A = \{1, 0\}$ . Dans ce cas-là, un symbole est appelé *bit*.

Dans le cas binaire, la distance de Hamming est donnée par l'expression suivante :

$$d_{\text{Hamming}}(a, b) = \sum_{i=0}^{n-1} (a_i \oplus b_i)$$

Où  $a_i$  et  $b_i$  sont les  $i^{\text{èmes}}$  bits des vecteurs binaires  $a$  et  $b$  respectivement,  $n$  est le nombre de bits ( $a$  et  $b$  ont la même longueur).  $\oplus$  représente l'opération binaire « ou exclusif » (XOR), qui renvoie 1 quand  $a_i$  et  $b_i$  sont différents et 0 quand ils ont la même valeur.

Dans C++, l'opérateur XOR représenté par le symbole  $\wedge$  reçoit deux entiers qui sont considérés dans leur représentation binaire. Le résultat est aussi un entier. Pour programmer la distance de Hamming entre la représentation binaire de deux entiers, il suffit de compter le nombre de bits à 1 dans le résultat du XOR. Le comptage de bits à 1 peut être programmé à partir d'autres opérations basiques. Dans l'encadré ci-dessous nous présentons la classe BitCounter qui implémente le comptage de bits à 1 pour différents des variables du type *long* qui décompose la variable de type *long* en sous parties de taille *char* et utilise une *lookup table* pour déterminer le nombre de bits à 1 de chaque sous-partie. La complexité du

comptage de bits et  $O(1)$ , la complexité est indépendante du nombre de bits dans le vecteur binaire.

```
/**
 *   Union allowing transformation between long long, long, char and unsigned int
 */
typedef union {
    long long jl;
    long ji[2];
    char jc[8];
    unsigned char uc[8];
    unsigned int ui[2];
} conversionUnion;

/**
 *   this class allows to count the number of bits to 1 in an integer represented as long long.
 *   it decomposes the integer into char memory blocks and retrieves the number of bits to
 *   1 in each block by accessing a precalculated lookup table represented by the array
 *   bitsCount
 */
class _BitCounter {
    /**
     *   Single instance of the class. This class implements the singleton software
     *   pattern
     */
    static _BitCounter* instance;
    /**
     *   Union used to convert between types
     */
    conversionUnion *converter;
    /**
     *   array containing the number of bits to 1 of every character value (between 0
     *   and 256)
     */
    int *bitsCount;
    /**
     *   The constructor and destructor are private to allow singleton implementation
     */
    _BitCounter();
    ~_BitCounter();
public :
    /**
```

```

    *      Creates the single instance of the class
    */
static void createInstance();
/**
    *      Destroys the single instance of the class
    */
static void destroyInstance();
/**
    *      Returns the number of bits to 1 of value
    */
static long countBits(long long value);
};
typedef _BitCounter * BitCounter;
_BitCounter* _BitCounter::instance = NULL;
conversionUnion *converter;
_BitCounter::_BitCounter(){
    converter = (conversionUnion *)malloc(sizeof(conversionUnion));
    bitsCount = (int *)malloc(sizeof(int)*256);
    //Fills up the lookup table
    for(int i=0; i<256; i++){
        bitsCount[i] = bitCount((long)i);
    }
}
_BitCounter::~_BitCounter(){
    free(converter);
    free(bitsCount);
}
void _BitCounter::createInstance(){
    if(_BitCounter::instance == NULL)
        _BitCounter::instance = new _BitCounter();
}
void _BitCounter::destroyInstance(){
    if(_BitCounter::instance != NULL){
        delete _BitCounter::instance;
        _BitCounter::instance =NULL;
    }
}
long _BitCounter::countBits(long long value){
    //Copies the value into the converter union

```

```

    _BitCounter::instance->converter->jl = value;
    //returns the sum of the bits to 1 of every char memory part of the value variable
    return instance->bitsCount[instance->converter->uc[0]] + instance->bitsCount[instance->converter-
>uc[1]] + instance->bitsCount[instance->converter->uc[2]] + instance->bitsCount[instance->converter->uc[3]]
+ instance->bitsCount[instance->converter->uc[4]] + instance->bitsCount[instance->converter->uc[5]] +
instance->bitsCount[instance->converter->uc[6]] + instance->bitsCount[instance->converter->uc[7]];
}

```

Encadré 26. Classe BitCounter qui implémente le comptage de bits à 1 dans un entier représenté sous forme de *long long* en C++

De cette manière nous pouvons calculer la distance de Hamming entre profils représentés par des entiers (voir Annexe II).

### A3.2. Distance Euclidienne

La distance euclidienne classique entre deux vecteurs de la même longueur est donnée par l'expression suivante :

$$d_{\text{Euclidienne}}(a, b) = \sqrt{\sum_{i=0}^{n-1} (a_i - b_i)^2}$$

Où  $a_i$  et  $b_i$  sont les  $i^{\text{èmes}}$  composantes des vecteurs  $a$  et  $b$ .

Lorsque l'on considère des vecteurs binaires, la terme  $(a_i - b_i)^2$  de l'expression ne peut valoir que 0 ou 1. En effet, étant donné que  $a_i$  et  $b_i$  ne peuvent valoir que 0 ou 1 il y a quatre possibles combinaisons :

- $a_i$  et  $b_i$  valent 1 tous les deux
- $a_i$  et  $b_i$  valent 0 tous les deux
- $a_i$  vaut 1 et  $b_i$  vaut 0
- $a_i$  vaut 0 et  $b_i$  vaut 1

Dans les deux premiers cas, le terme  $(a_i - b_i)^2$  vaut 0. Dans les deux deuxièmes cas, ce terme vaut 1. Autrement dit, lorsque  $a_i$  est égal à  $b_i$ ,  $(a_i - b_i)^2$  vaut 0 et lorsque  $a_i$  est différent de  $b_i$ ,  $(a_i - b_i)^2$  vaut 1. C'est la définition de l'opération XOR. On peut donc ré-écrire l'expression de la distance euclidienne pour le cas des vecteurs binaires comme il suit :

$$d_{\text{Euclidienne}}(a, b) = \sqrt{\sum_{i=0}^{n-1} (a_i \oplus b_i)}$$

Le terme  $\sum_{i=0}^{n-1} (a_i \oplus b_i)$  à l'intérieur de la racine carrée la distance de Hamming. Donc, on peut en déduire que pour le cas de vecteurs binaires la distance euclidienne dépend de la distance de Hamming selon l'expression suivante :

$$d_{\text{Euclidienne}}(a, b) = \sqrt{d_{\text{Hamming}}(a, b)}$$

Le calcul de la distance euclidienne par l'expression ci-dessous n'ajoute qu'une opération au calcul de la distance de Hamming, la complexité de ce calcul est donc indépendante du nombre de modalités dans les profils. La complexité de ce calcul est  $O(1)$ .



## Annexe IV. Mesures de corrélation de deux ensembles de données

Dans cette annexe, nous présentons plusieurs mesures de corrélation entre deux ensembles de données numériques. Nous utilisons ces mesures pour valider les distributions de populations simulées par rapport aux distributions de populations réelles. Nous nous intéressons uniquement aux mesures croisées qui donnent une valeur normalisée de corrélation entre deux ensembles de données. En effet, il existe d'autres mesures statistiques, par exemple la variance des différences entre les deux ensembles. Le problème de ces mesures est qu'elles ne sont pas bornées et ne peuvent servir que dans un cadre comparatif entre plusieurs validations. Les mesures normalisées sont plus faciles d'interprétation et peuvent être utilisées directement car elles donnent une qualification globale de chaque validation.

Nous présentons trois mesures de corrélation normalisée : la corrélation croisée normalisée, la corrélation croisée normalisée et centrée ainsi que la corrélation de Moravec. Ces trois mesures sont basées sur la corrélation croisée (ou produit de corrélation) donnée par :

$$CC(ds, dr) = ds \cdot dr$$

où  $ds$  est le vecteur de valeurs à comparer et  $dr$  le vecteur de référence,  $ds \cdot dr$  est le produit scalaire entre les deux vecteurs de données, c'est-à-dire :

$$ds \cdot dr = \sum_{i=0}^n (ds_i \times dr_i)$$

où  $n$  est le nombre de composants des vecteurs  $ds$  et  $dr$  et les valeurs  $ds_i$  et  $dr_i$  sont les  $i$ èmes composantes des vecteurs  $ds$  et  $dr$  respectivement.

L'expression de la corrélation croisée n'est pas bornée et ne peut pas être utilisée directement car plus les valeurs absolues des données dans les vecteurs sont élevées, plus la valeur de  $CC$  est élevée. En effet, les valeurs de  $CC(ds, dr) = ds \cdot dr$  appartiennent à l'intervalle  $[-\infty; +\infty]$ .

### A4.1. Corrélation croisée normalisée

La mesure, notée  $NCC$  (Normalized Cross-Correlation) est donnée par :

$$NCC(ds, dr) = \frac{ds \cdot dr}{\|ds\| \times \|dr\|}$$

où  $\|ds\|$  et  $\|dr\|$  sont les normes euclidienne des vecteurs  $ds$  et  $dr$  respectivement. Ces normes sont calculées par l'expression suivante :

$$\|d\| = \sqrt{\sum_{i=0}^n d_i^2}$$

Les valeurs de  $NCC(ds, dr)$  appartiennent à l'intervalle  $[-1 ; 1]$

### A4.2. Corrélation croisée normalisée centrée

La mesure, notée  $ZNCC$  (Zero mean Normalized Cross-Correlation) est donnée par :

$$ZNCC(ds, dr) = \frac{(ds - \bar{ds}) \cdot (dr - \bar{dr})}{\|ds - \bar{ds}\| \times \|dr - \bar{dr}\|} = \frac{\sum_{i=0}^n (ds_i - \bar{ds}) \times (dr_i - \bar{dr})}{\sqrt{\left(\sum_{i=0}^n (ds_i - \bar{ds})^2\right)} \times \sqrt{\left(\sum_{i=0}^n (dr_i - \bar{dr})^2\right)}}$$

où  $\bar{ds}$  est la valeur moyenne du vecteur  $ds$  et  $\bar{dr}$  est la valeur moyenne du vecteur  $dr$ . Le  $ZNCC$  calcule la corrélation centrée autour des moyennes des vecteurs comparés. Les valeurs de  $ZNCC(ds, dr)$  appartiennent à l'intervalle  $[-1 ; 1]$

### A4.3 Corrélation de Moravec

La mesure de Moravec utilise une normalisation différente de celle utilisée par la mesure  $ZNCC$  :

$$MOR(ds, dr) = \frac{2 \times (ds - \overline{ds}) \cdot (dr - \overline{dr})}{\|ds - \overline{ds}\|^2 + \|dr - \overline{dr}\|^2}$$

Les valeurs de  $MOR(ds, dr)$  appartiennent à l'intervalle  $[-1 ; 1]$

## Annexe V. Exemples de résultats graphiques des simulations effectuées

Dans cette partie, nous présentons les résultats graphiques les plus importants pour l'arrondissement « La Candelaria » à Bogotá obtenus à partir des simulations avec le modèle étendu en utilisant l'algorithme de classification basé sur les nuages de particules, dans la formation des groupes (simulations du type ME-DM-PART, cf. section 11.2.4.). Dans ces simulations, nous avons considéré le paramétrage de référence.

### A5.1. Diagrammes historiques

Dans les figures ci-dessous, nous présentons les diagrammes historiques (cf. section 10.2.3) pour les groupes sociaux et les types d'habitat pour l'arrondissement « La Candelaria » obtenus par l'exécution des simulations du type ME-DM-PART avec le paramétrage de référence. Dans ces simulations, 12 groupes sociaux et 58 types d'habitat ont émergé. Dans l'année finale de simulation, il reste 10 groupes sociaux et seulement 11 types d'habitat dans l'année finale de simulation.

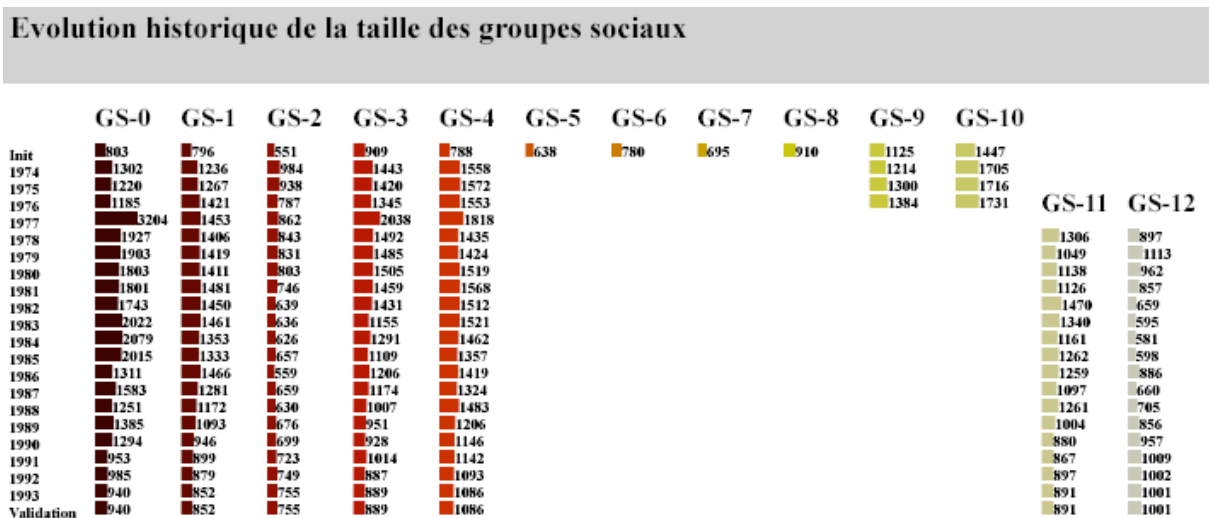


Figure 68. Diagramme historique des groupes sociaux pour l'arrondissement « La Candelaria » à Bogotá. Résultats des simulations du type ME-DM-PART avec le paramétrage de référence.

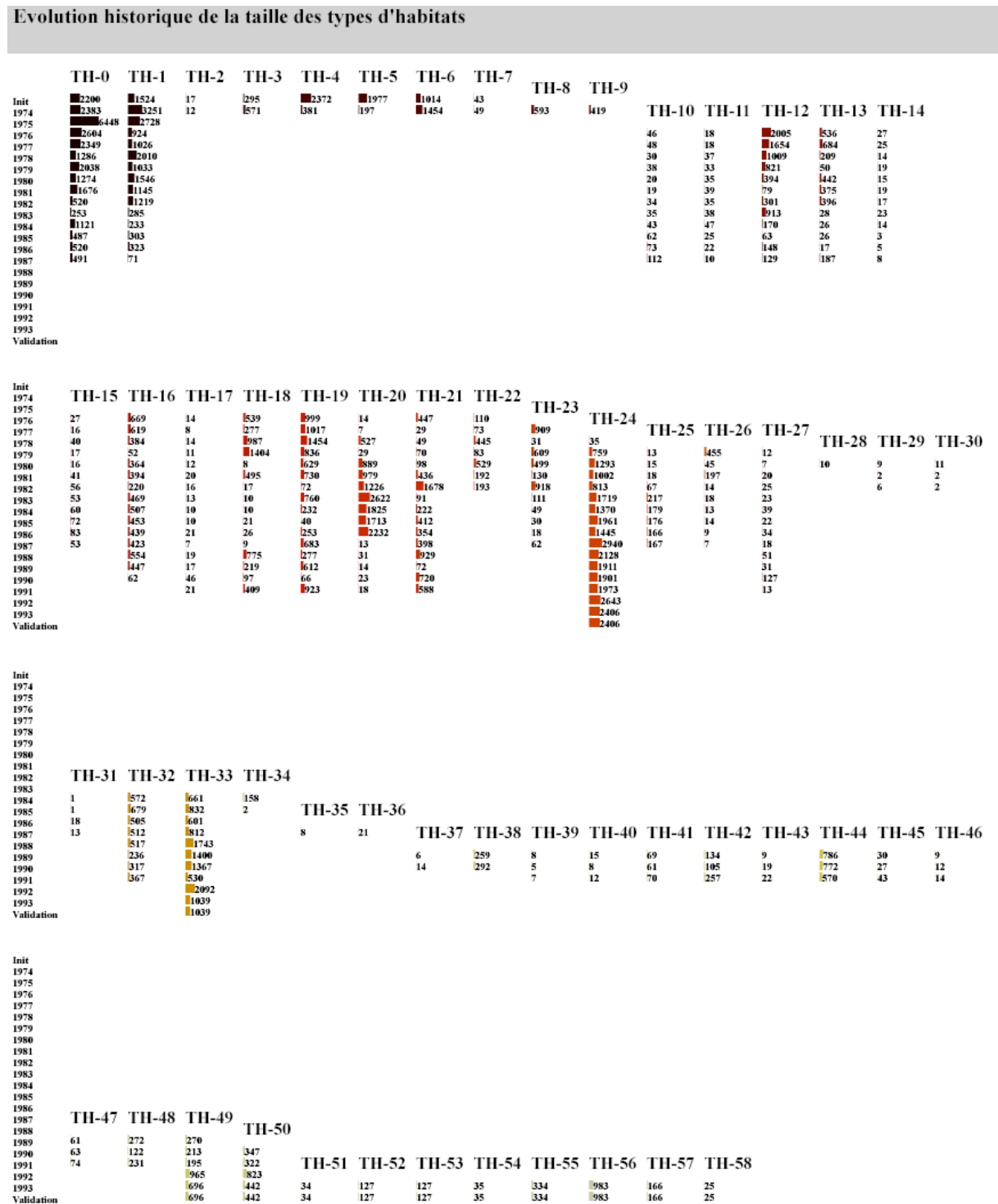


Figure 69. Diagramme historique des types d'habitats pour l'arrondissement « La Candelaria » à Bogotá. Résultats des simulations du type ME-DM-PART avec le paramétrage de référence.

## A5.2. Diagrammes de caractérisation automatique des groupes

Dans cette section, nous présentons les diagrammes de caractérisation automatique (diagrammes de valeur-test, cf. section 10.2.2.) des groupes sociaux et des types d'habitat de l'année finale de simulation.

### A5.2.1. Diagrammes de caractérisation automatique des groupes sociaux

Caractérisation automatique du groupe GS-0 pour l'année 1993 (population simulée)  
Selon les valeurs de test

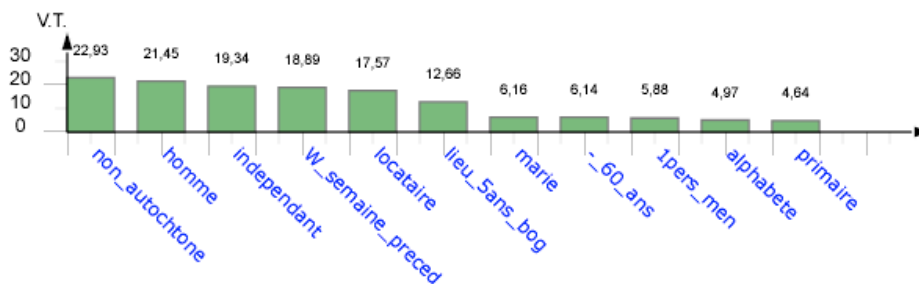


Figure 70. Diagramme de description automatique (valeurs test) du groupe social GS-0.

Caractérisation automatique du groupe GS-1 pour l'année 1993 (population simulée)  
Selon les valeurs de test

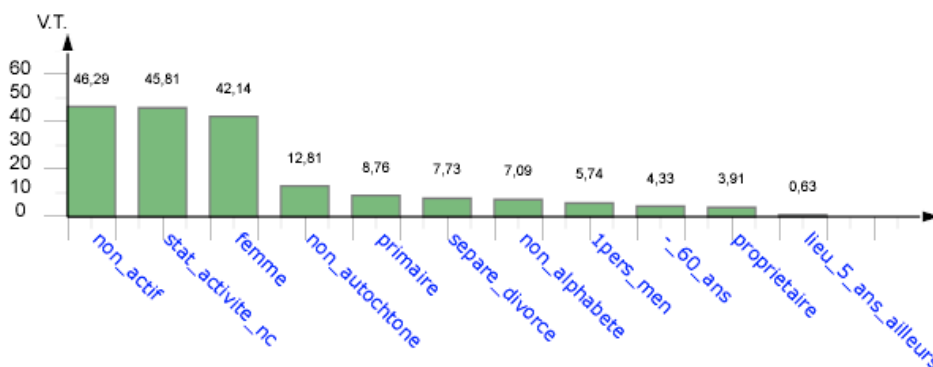


Figure 71. Diagramme de description automatique (valeurs test) du groupe social GS-1.

Caractérisation automatique du groupe GS-2 pour l'année 1993 (population simulée)  
Selon les valeurs de test

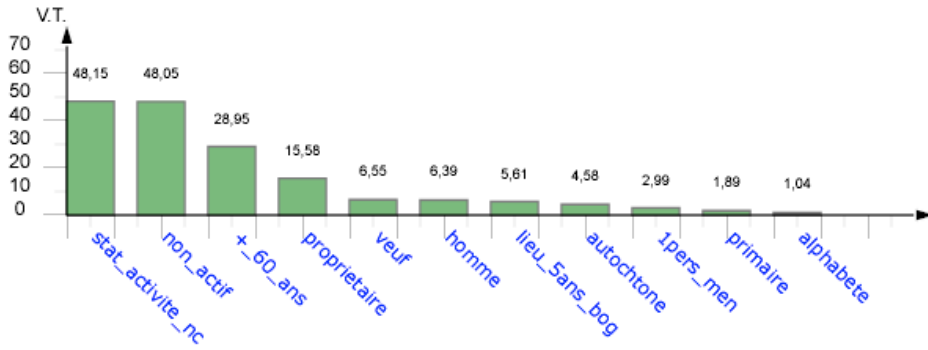


Figure 72. Diagramme de description automatique (valeurs test) du groupe social GS-2.

Caractérisation automatique du groupe GS-3 pour l'année 1993 (population simulée)  
Selon les valeurs de test

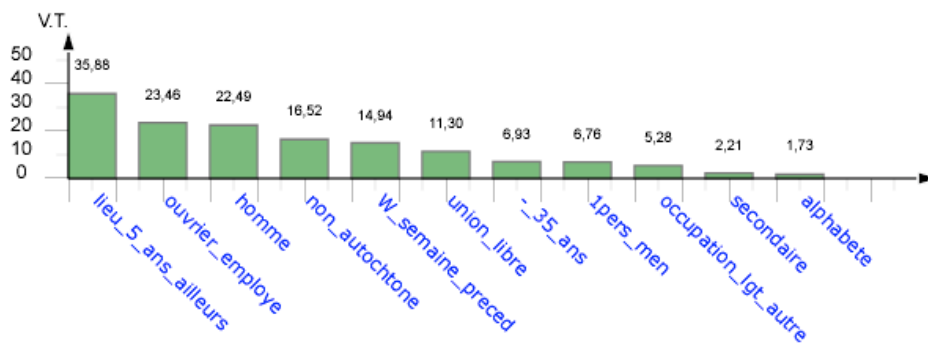


Figure 73. Diagramme de description automatique (valeurs test) du groupe social GS-3.

Caractérisation automatique du groupe GS-11 pour l'année 1993 (population simulée)  
Selon les valeurs de test

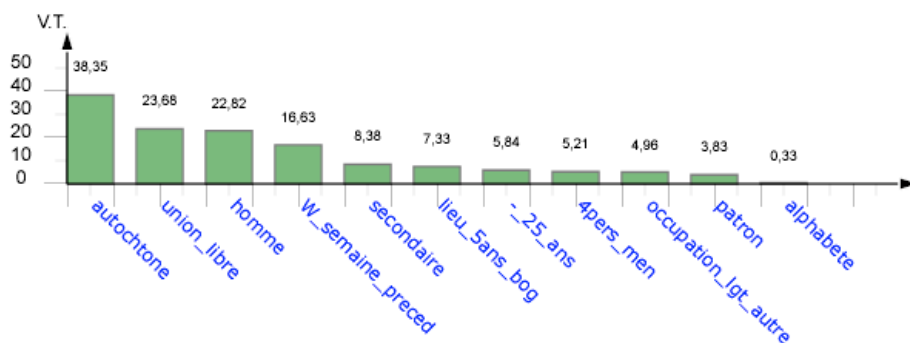


Figure 74. Diagramme de description automatique (valeurs test) du groupe social GS-11.

Caractérisation automatique du groupe GS-12 pour l'année 1993 (population simulée)  
Selon les valeurs de test

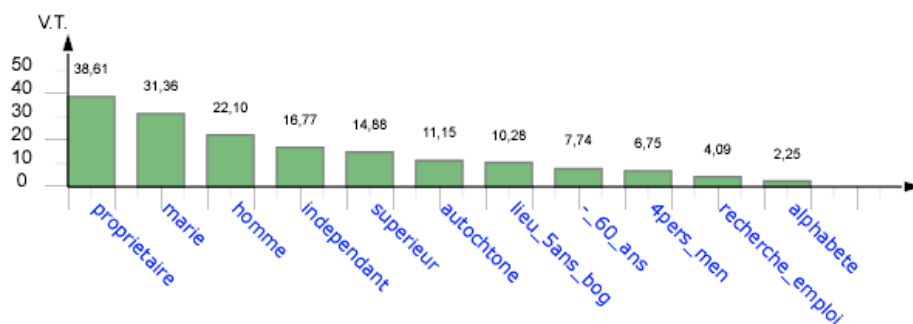


Figure 75. Diagramme de description automatique (valeurs test) du groupe social GS-12.



## A5.2.2. Caractérisation automatique des types d'habitats

Caractérisation automatique du groupe TH-24 pour l'année 1993 (population simulée)  
Selon les valeurs de test

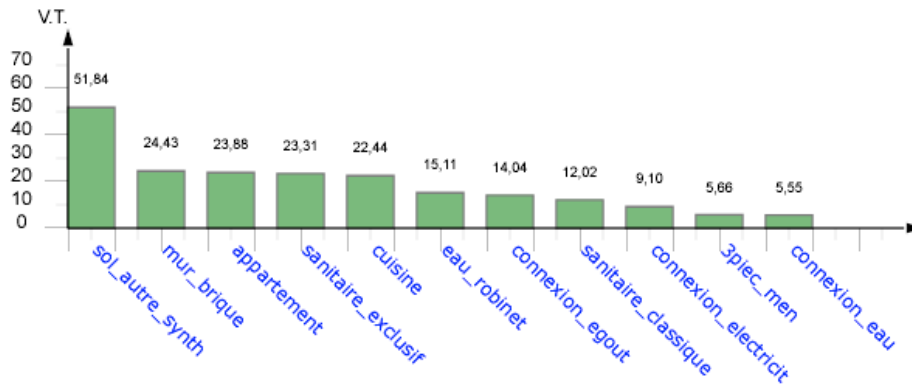


Figure 76. Diagramme de description automatique (valeurs test) du type d'habitat TH-24.

Caractérisation automatique du groupe TH-33 pour l'année 1993 (population simulée)  
Selon les valeurs de test

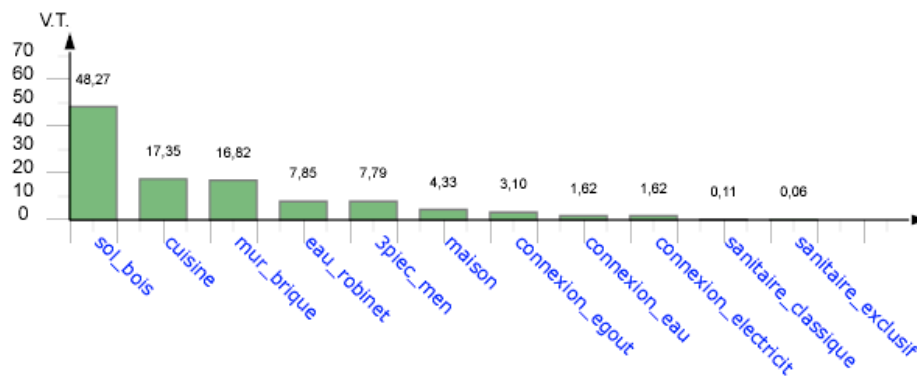


Figure 77. Diagramme de description automatique (valeurs test) du type d'habitat TH-33.

Caractérisation automatique du groupe TH-49 pour l'année 1993 (population simulée)  
Selon les valeurs de test

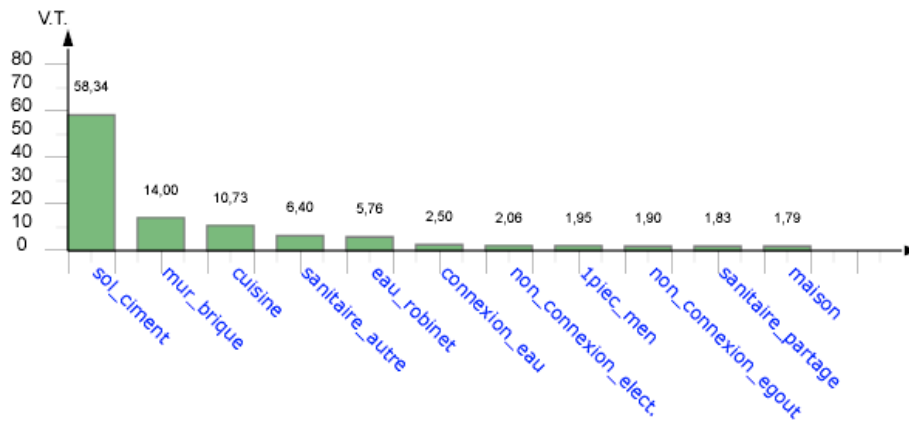


Figure 78. Diagramme de description automatique (valeurs test) du type d'habitat TH-49.

Caractérisation automatique du groupe TH-50 pour l'année 1993 (population simulée)  
Selon les valeurs de test

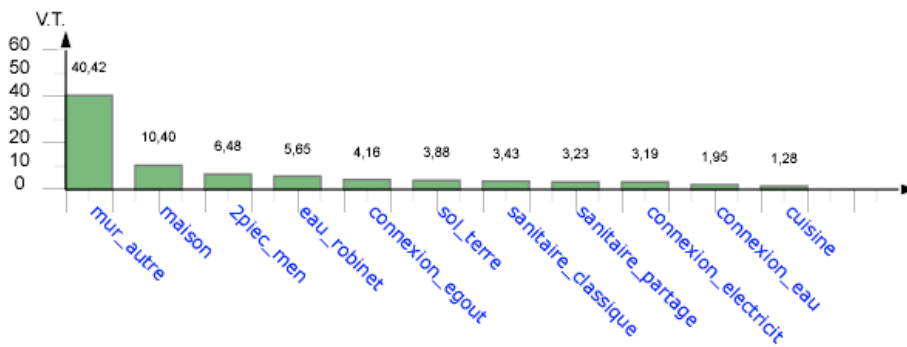


Figure 79. Diagramme de description automatique (valeurs test) du type d'habitat TH-50.

Caractérisation automatique du groupe TH-51 pour l'année 1993 (population simulée)  
Selon les valeurs de test

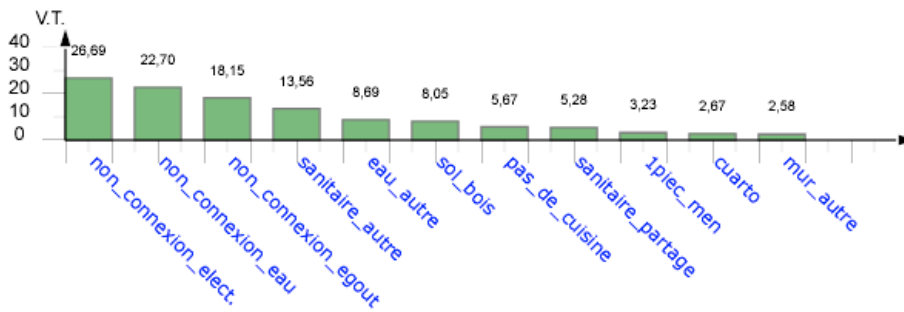


Figure 80. Diagramme de description automatique (valeurs test) du type d'habitat TH-51.

Caractérisation automatique du groupe TH-52 pour l'année 1993 (population simulée)  
Selon les valeurs de test

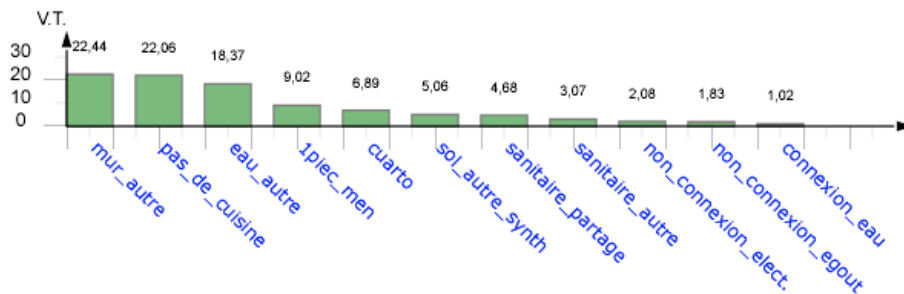


Figure 81. Diagramme de description automatique (valeurs test) du type d'habitat TH-52.

Caractérisation automatique du groupe TH-53 pour l'année 1993 (population simulée)  
Selon les valeurs de test

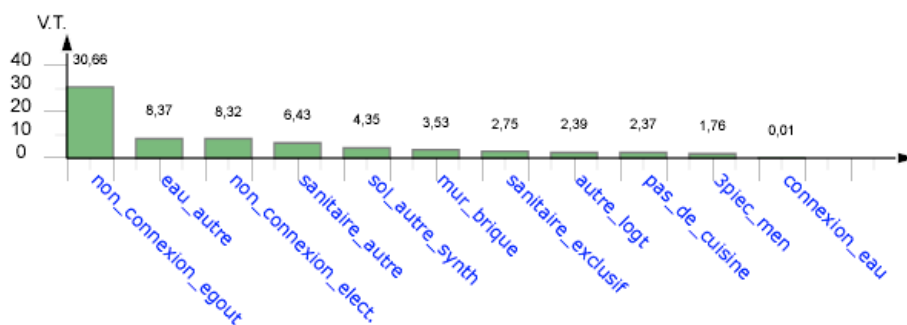


Figure 82. Diagramme de description automatique (valeurs test) du type d'habitat TH-53.

Caractérisation automatique du groupe TH-54 pour l'année 1993 (population simulée)  
Selon les valeurs de test

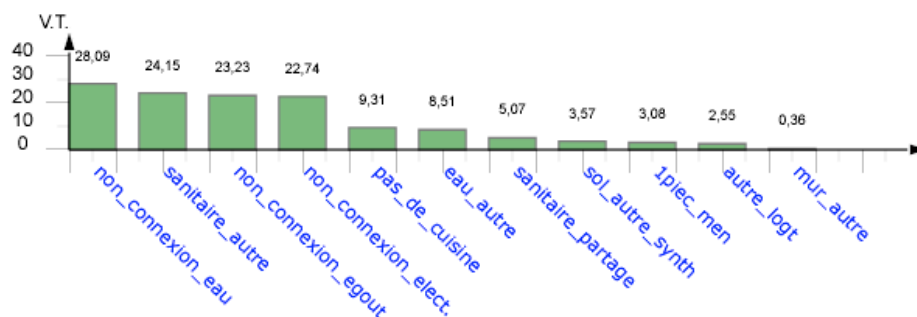


Figure 83. Diagramme de description automatique (valeurs test) du type d'habitat TH-54.

Caractérisation automatique du groupe TH-55 pour l'année 1993 (population simulée)  
Selon les valeurs de test

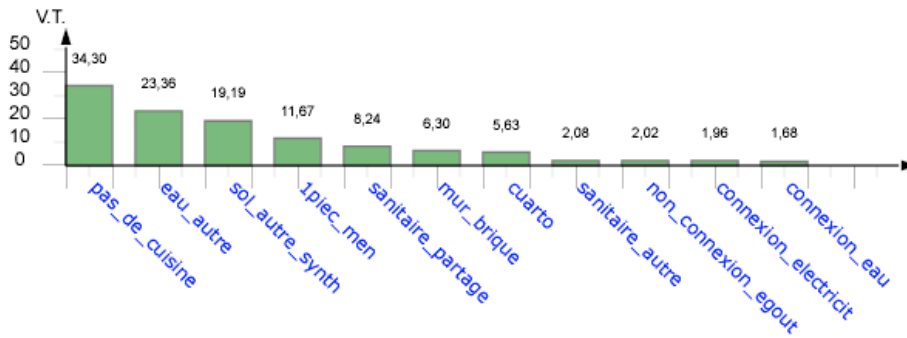


Figure 84. Diagramme de description automatique (valeurs test) du type d'habitat TH-55.

Caractérisation automatique du groupe TH-56 pour l'année 1993 (population simulée)  
Selon les valeurs de test

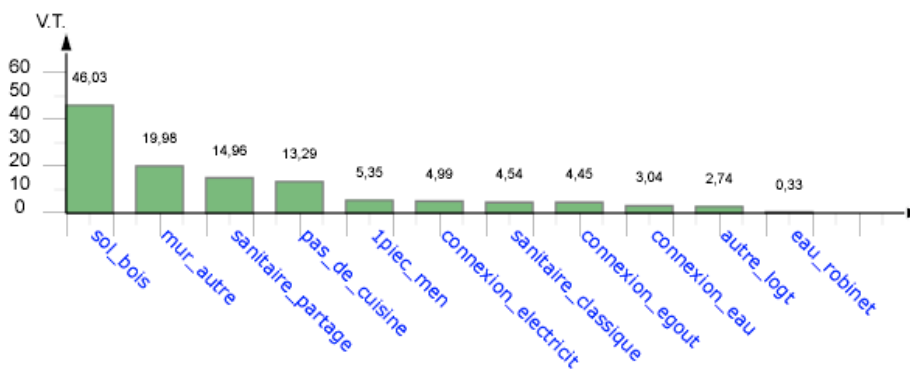


Figure 85. Diagramme de description automatique (valeurs test) du type d'habitat TH-56.

Caractérisation automatique du groupe TH-57 pour l'année 1993 (population simulée)  
Selon les valeurs de test

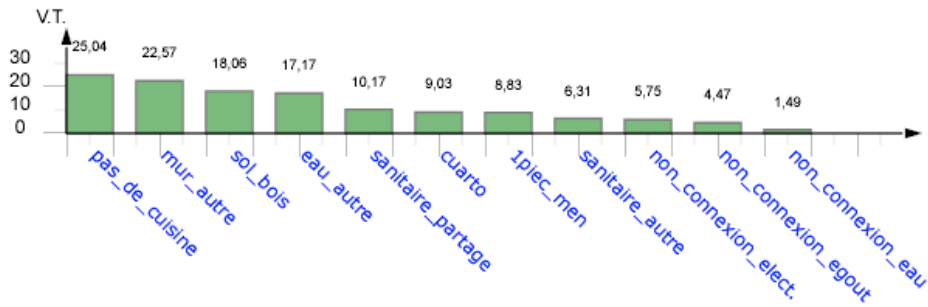


Figure 86. Diagramme de description automatique (valeurs test) du type d'habitat TH-57.

Caractérisation automatique du groupe TH-58 pour l'année 1993 (population simulée)  
Selon les valeurs de test

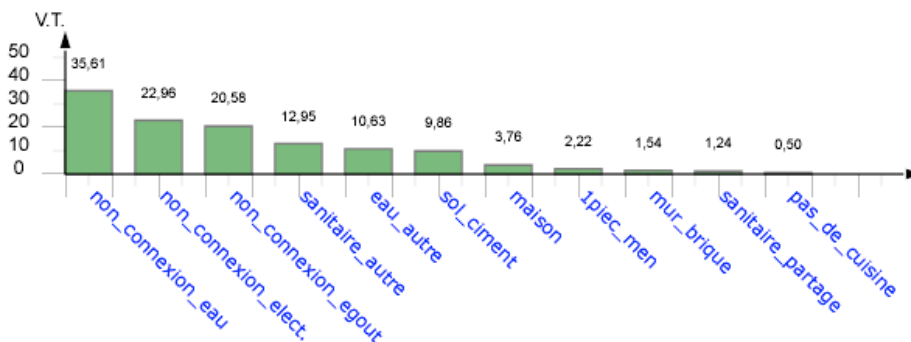


Figure 87. Diagramme de description automatique (valeurs test) du type d'habitat TH-58.

### A5.3. Cartes de validation de la distribution des groupes sociaux et de l'occupation des types d'habitat

Dans cette section, nous présentons les cartes de validation de la distribution spatiale des groupes sociaux et les cartes de validation de l'occupation des types d'habitat (cf. section 9.2.)

pour l'arrondissement « La Candelaria ». Ces résultats ont été obtenus par des simulations du type ME-DM-PART avec le paramétrage de référence.

### A5.3.1. Cartes de validation de la distribution spatiale des groupes sociaux pour l'année finale de simulation

Dans les figures ci-dessous, nous présentons les cartes des validations pour l'arrondissement « La Candelaria ». Ces cartes ont été produites par la comparaison de la population simulée dans l'année finale de simulation avec la distribution réelle. Les partitions considérées pour les groupes sociaux et les types d'habitat sont les mêmes pour les groupes sociaux et les types d'habitats.

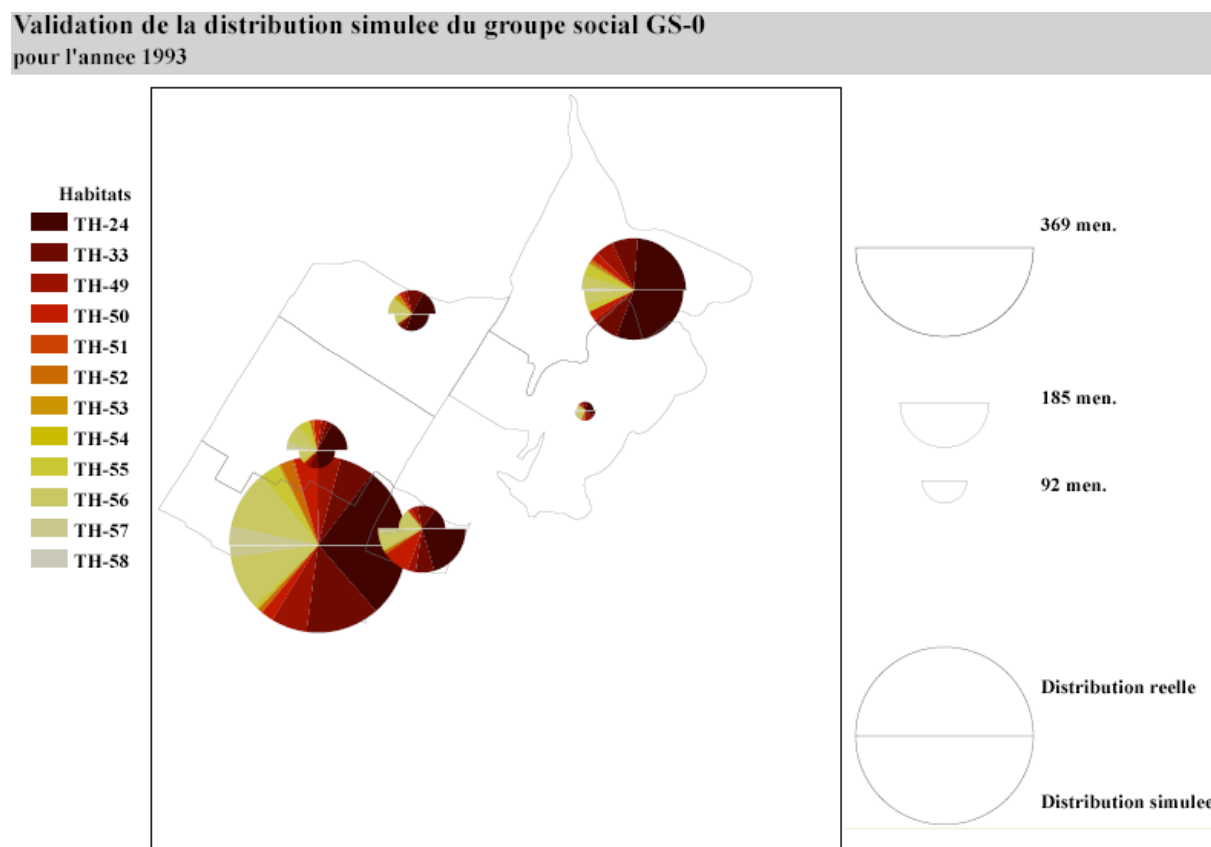


Figure 88. Carte de validation pour le groupe social GS-0.

**Validation de la distribution simulee du groupe social GS-1  
pour l'annee 1993**

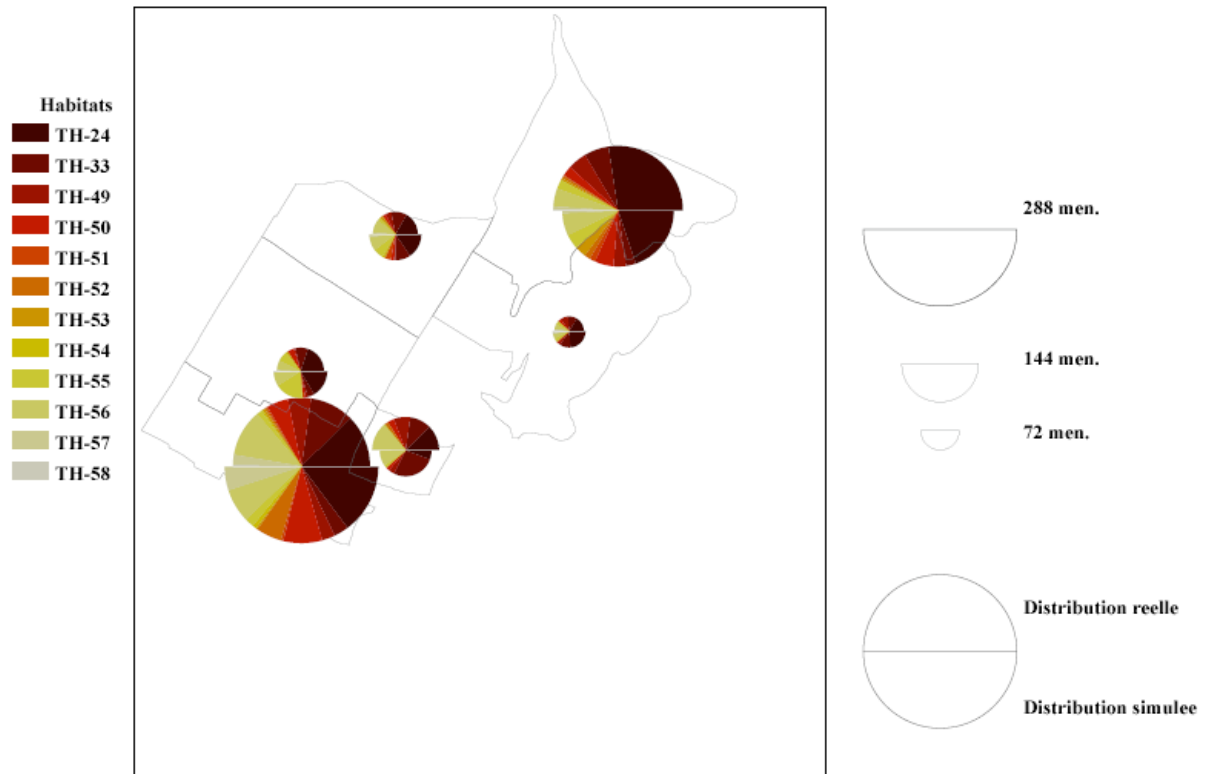


Figure 89. Carte de validation pour le groupe social GS-1.

**Validation de la distribution simulee du groupe social GS-2  
pour l'annee 1993**

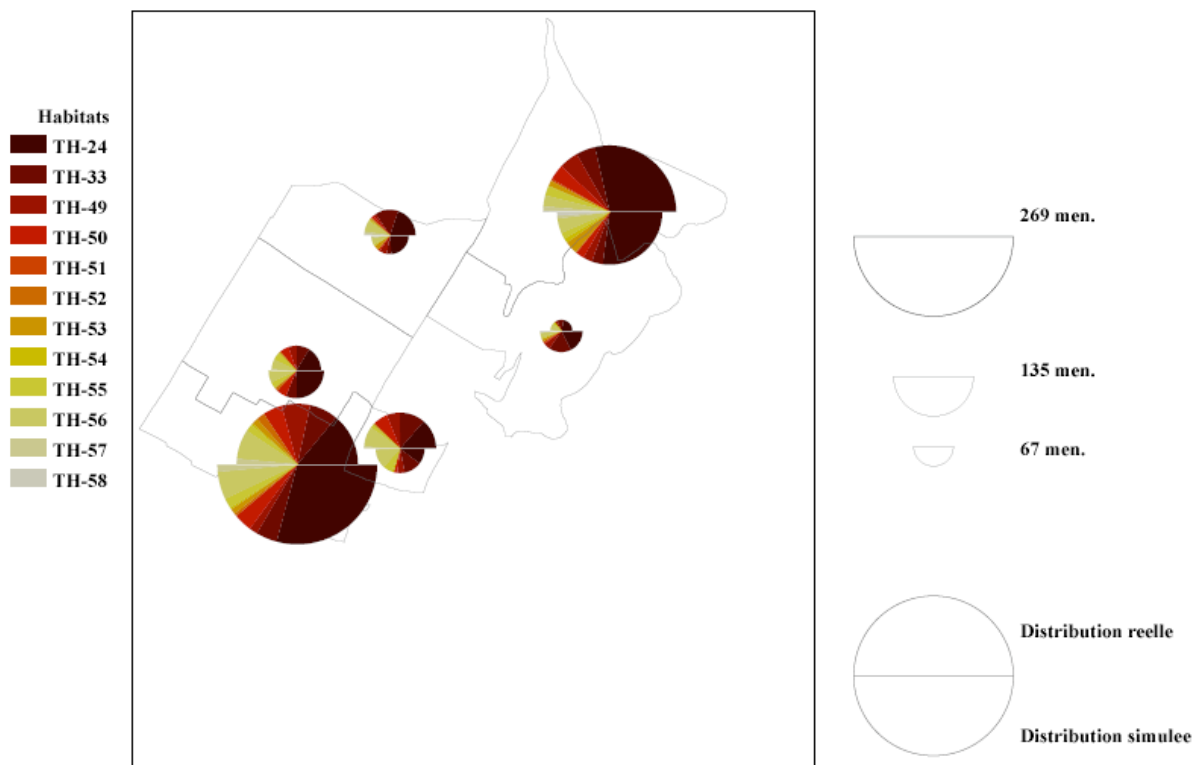


Figure 90. Carte de validation pour le groupe social GS-2.



**Validation de la distribution simulee du groupe social GS-3  
pour l'annee 1993**

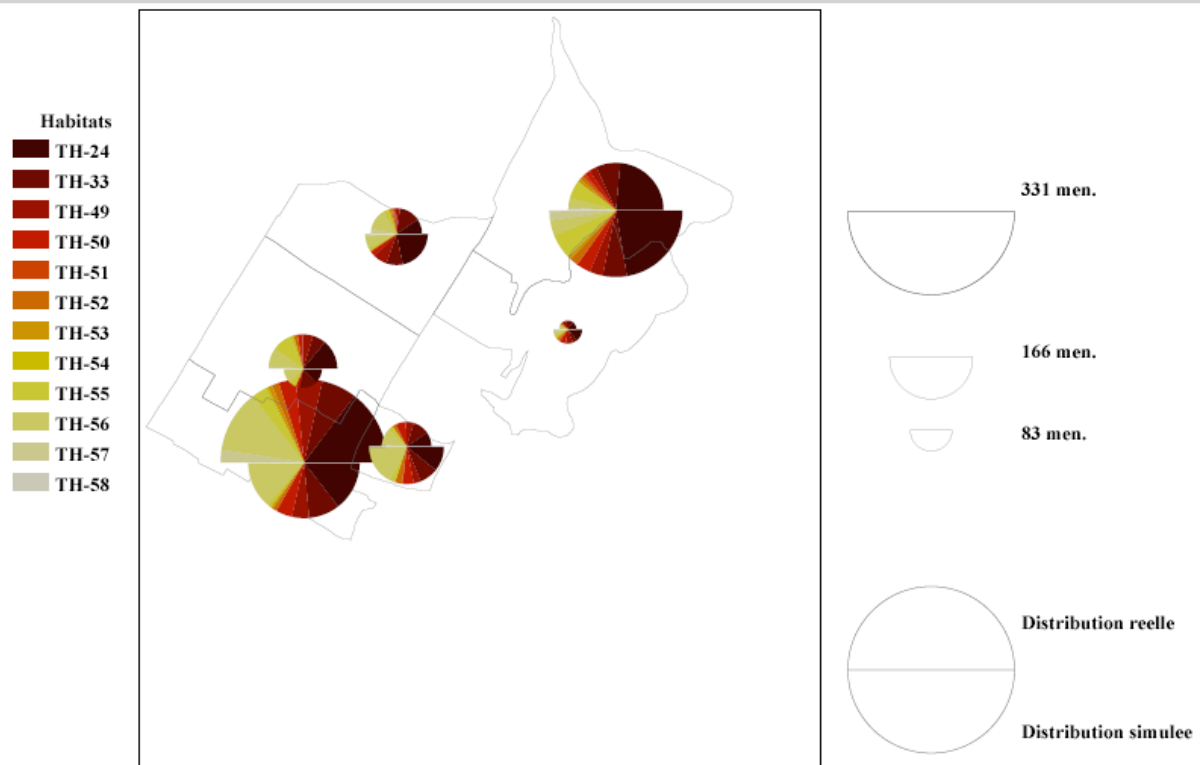


Figure 91. Carte de validation pour le groupe social GS-3.

**Validation de la distribution simulee du groupe social GS-4  
pour l'annee 1993**

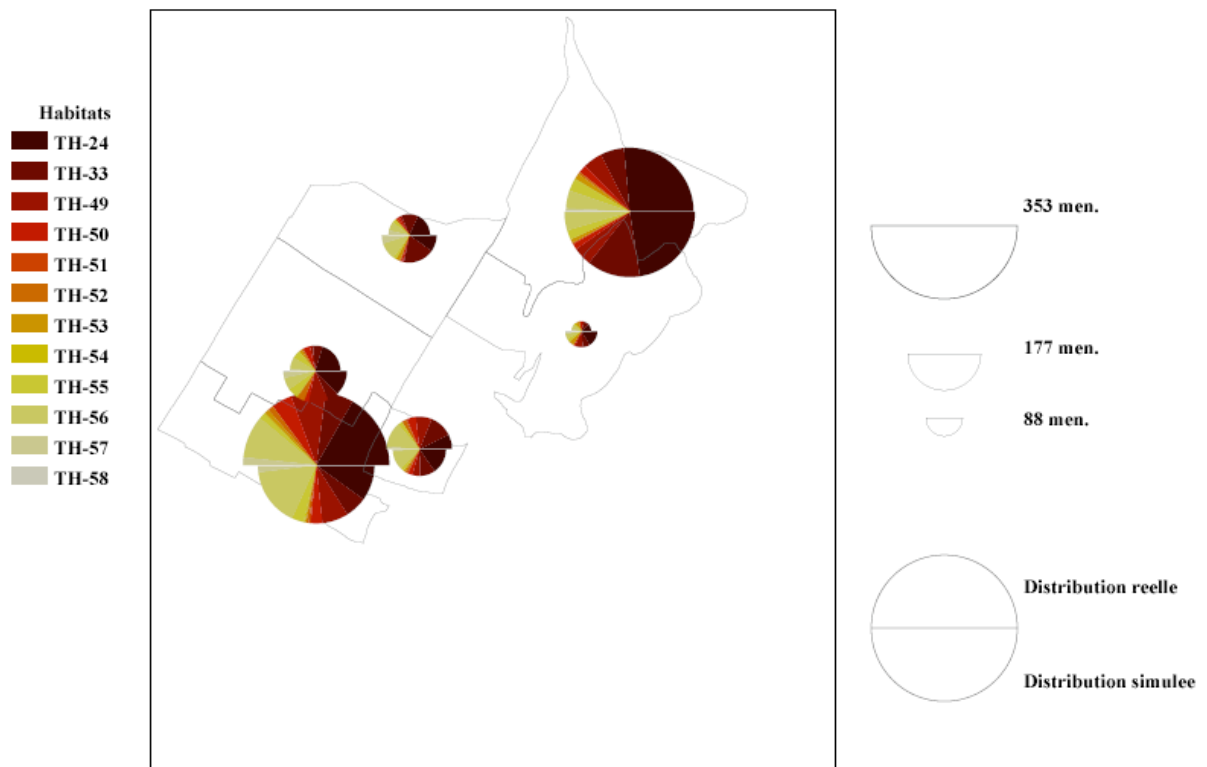


Figure 92. Carte de validation pour le groupe social GS-4.

**Validation de la distribution simulee du groupe social GS-11  
pour l'annee 1993**

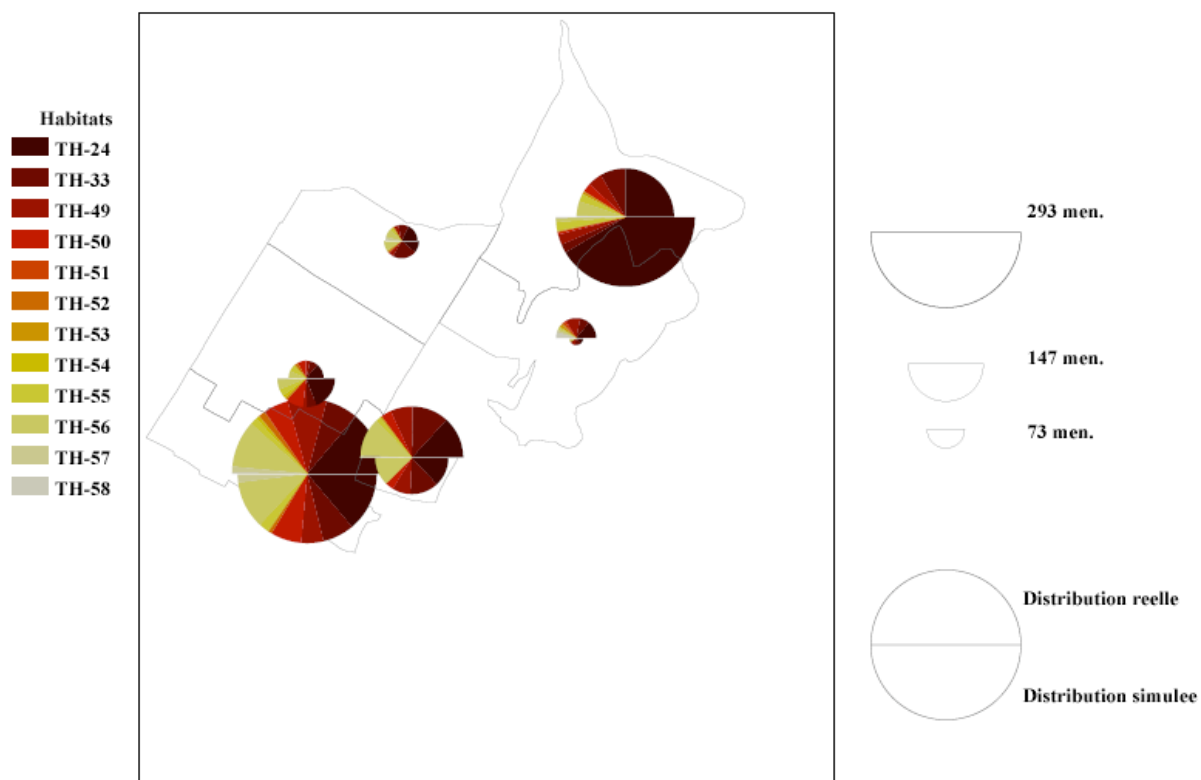


Figure 93. Carte de validation pour le groupe social GS-11.

**Validation de la distribution simulee du groupe social GS-12  
pour l'annee 1993**

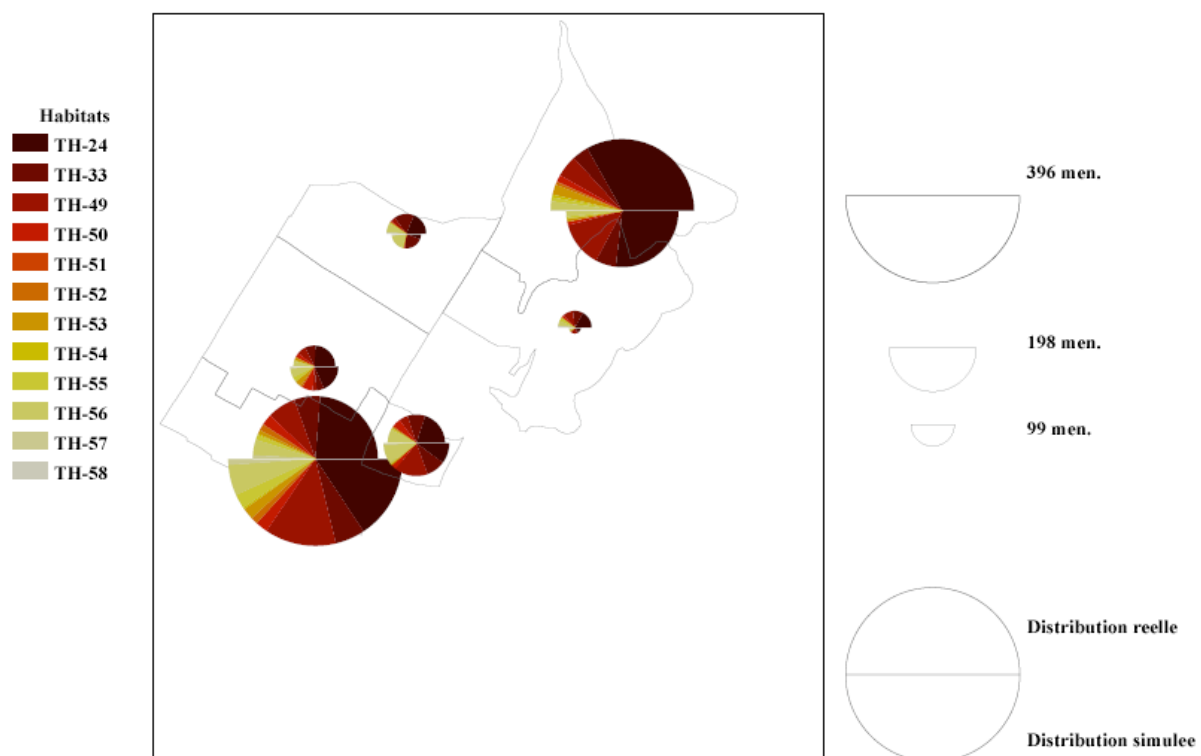


Figure 94. Carte de validation pour le groupe social GS-12.

### A5.3.1. Cartes de validation de l'occupation des types d'habitat pour l'année finale de simulation

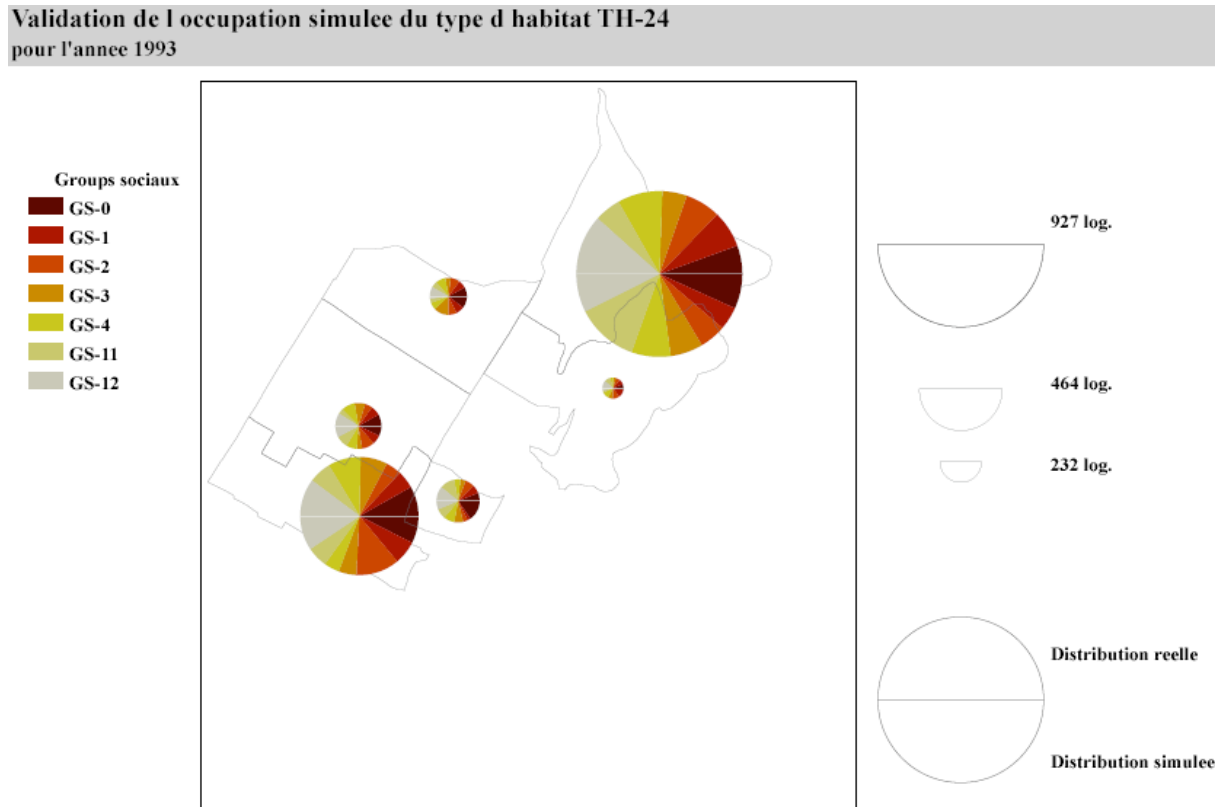


Figure 95. Carte de validation pour le type d'habitat TH-24.

**Validation de l'occupation simulee du type d'habitat TH-33  
pour l'annee 1993**

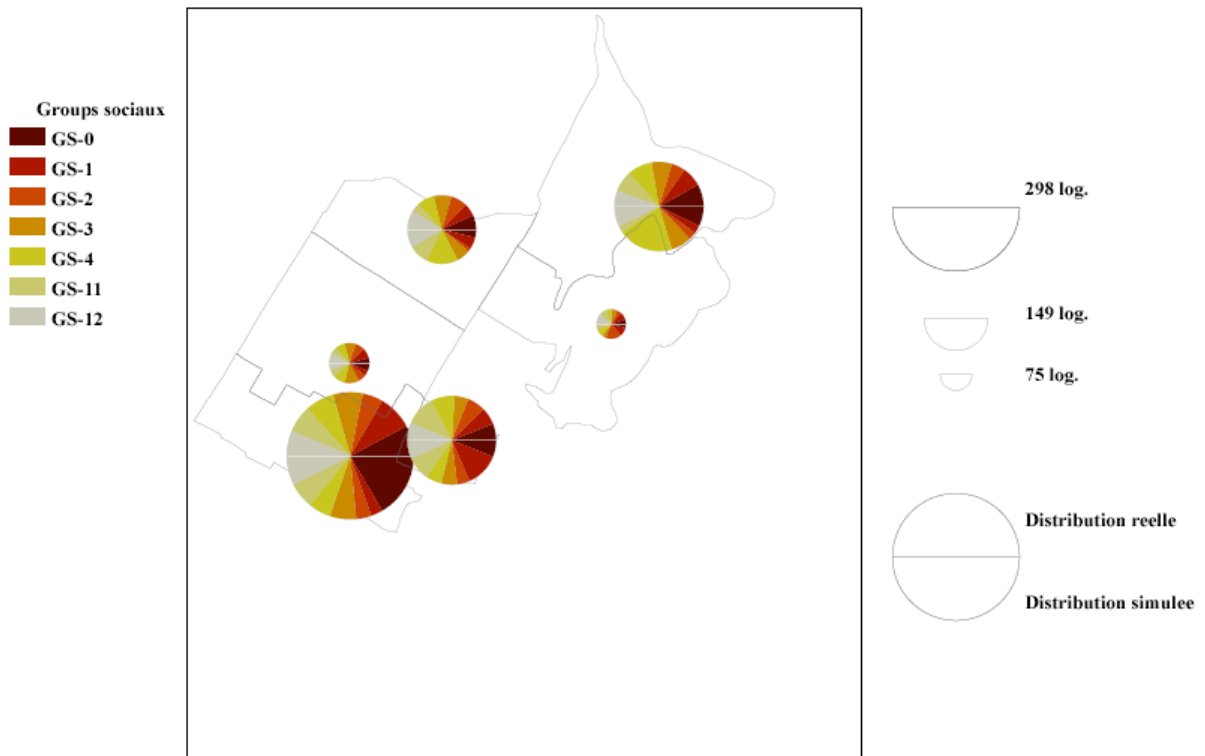


Figure 96. Carte de validation pour le type d'habitat TH-33.

**Validation de l'occupation simulee du type d'habitat TH-49  
pour l'annee 1993**

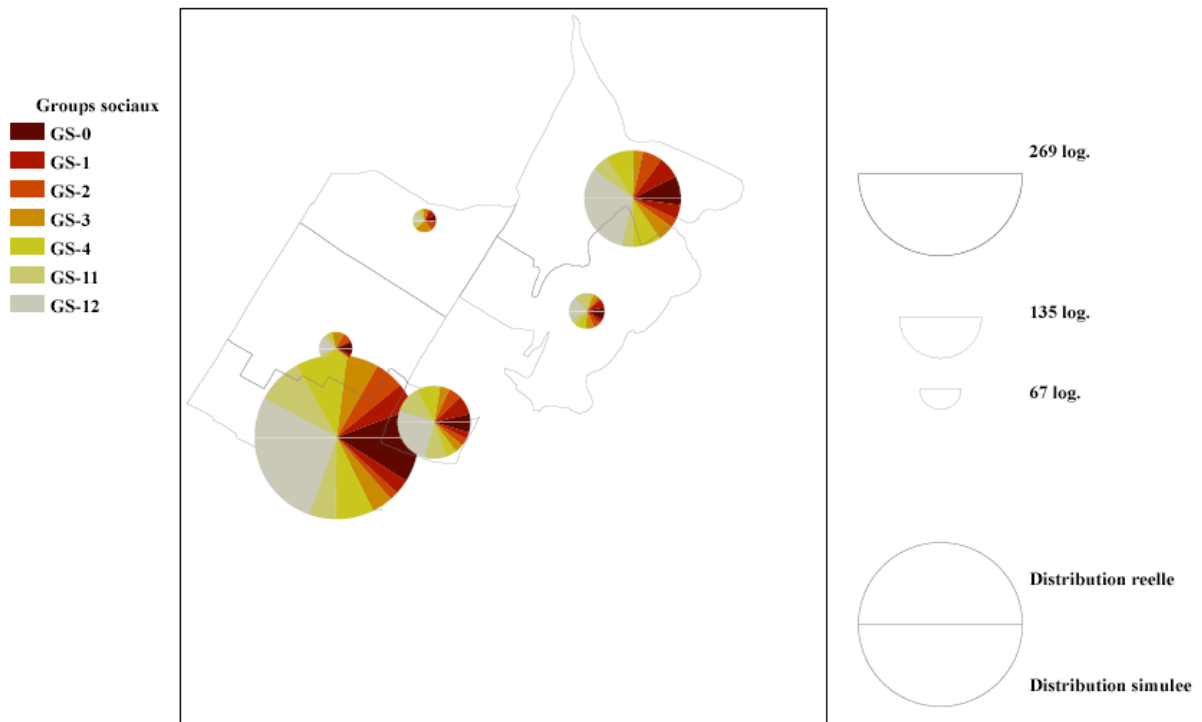


Figure 97. Carte de validation pour le type d'habitat TH-49.

**Validation de l'occupation simulee du type d'habitat TH-50  
pour l'annee 1993**

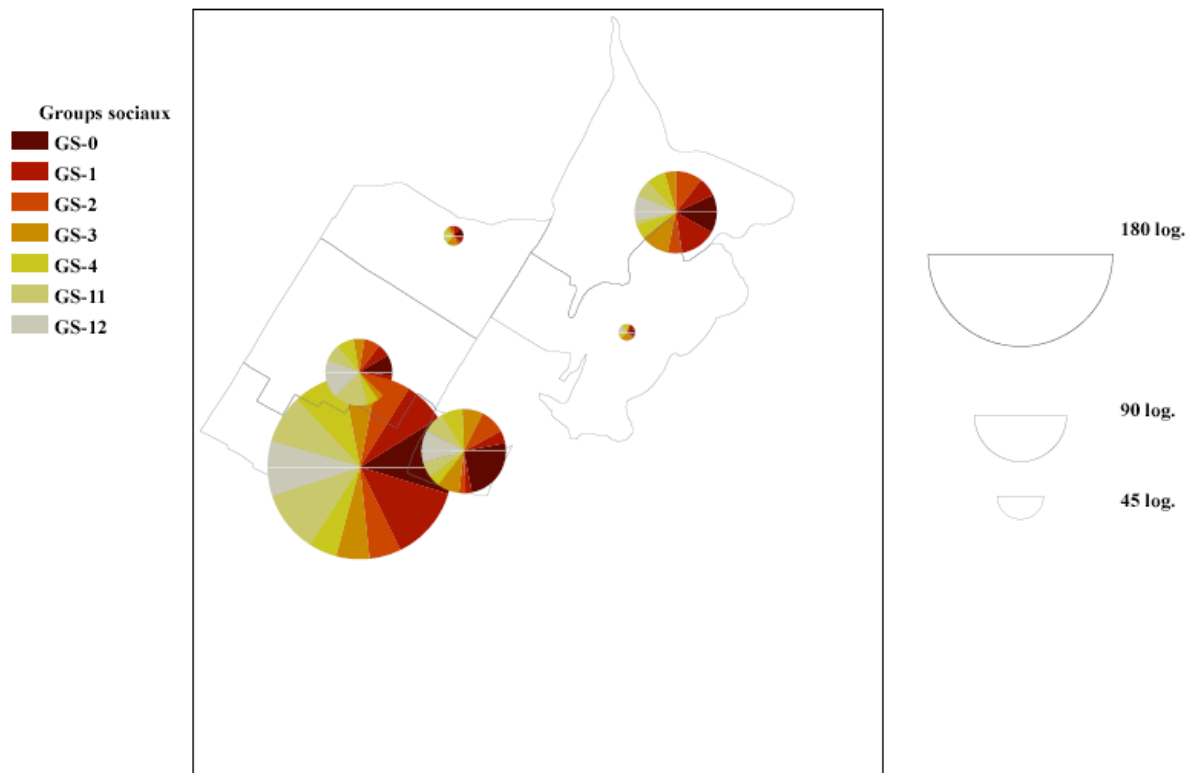


Figure 98. Carte de validation pour le type d'habitat TH-50.

**Validation de l'occupation simulee du type d'habitat TH-52  
pour l'annee 1993**

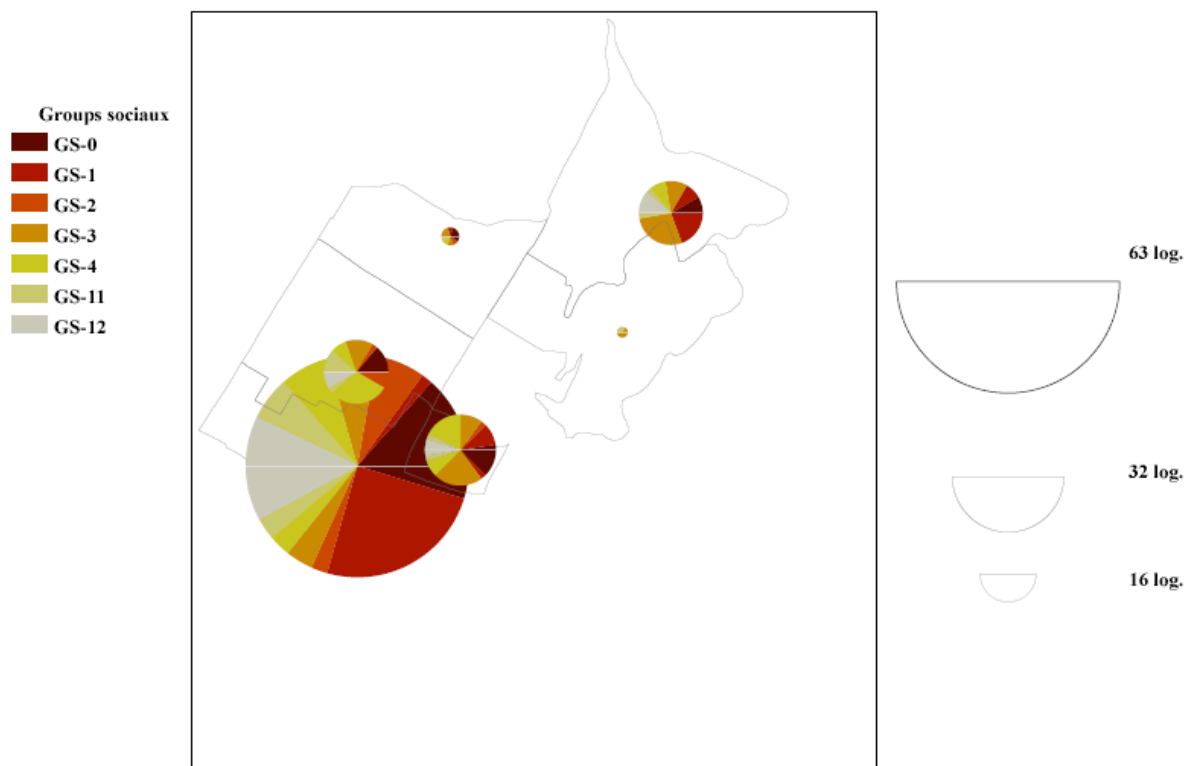


Figure 99. Carte de validation pour le type d'habitat TH-52.

**Validation de l'occupation simulee du type d'habitat TH-53  
pour l'annee 1993**

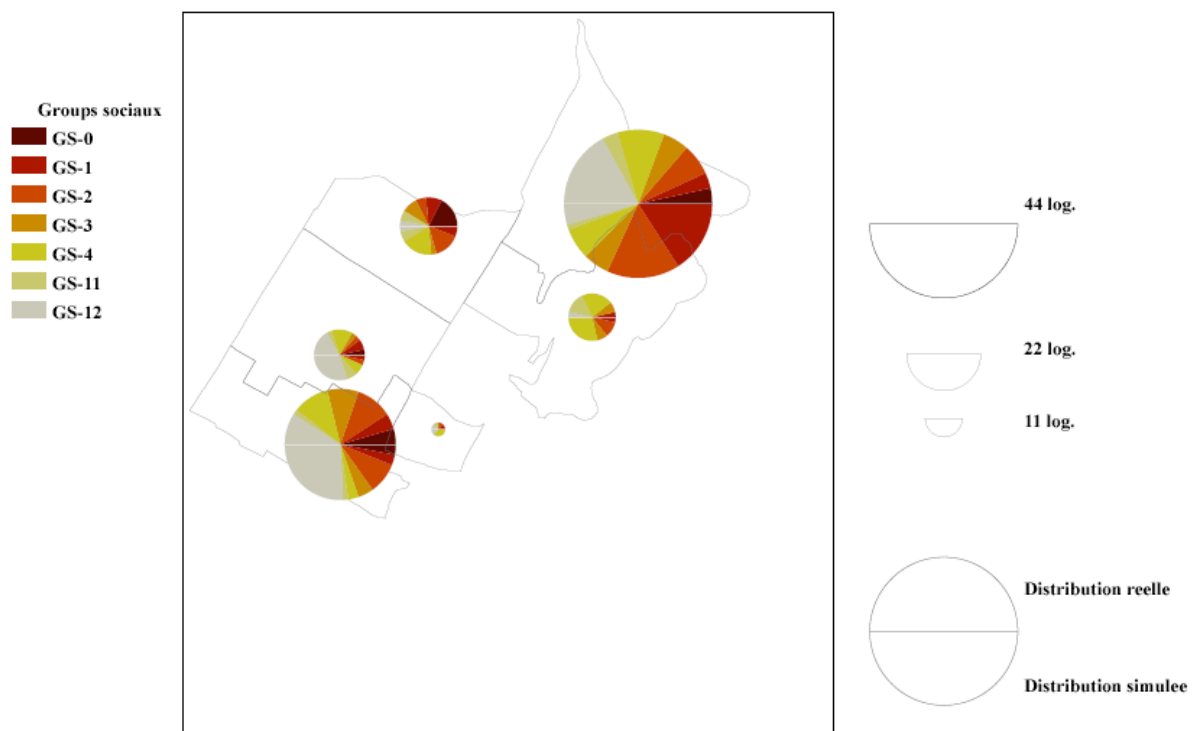


Figure 100. Carte de validation pour le type d'habitat TH-53.

**Validation de l'occupation simulee du type d'habitat TH-54  
pour l'annee 1993**

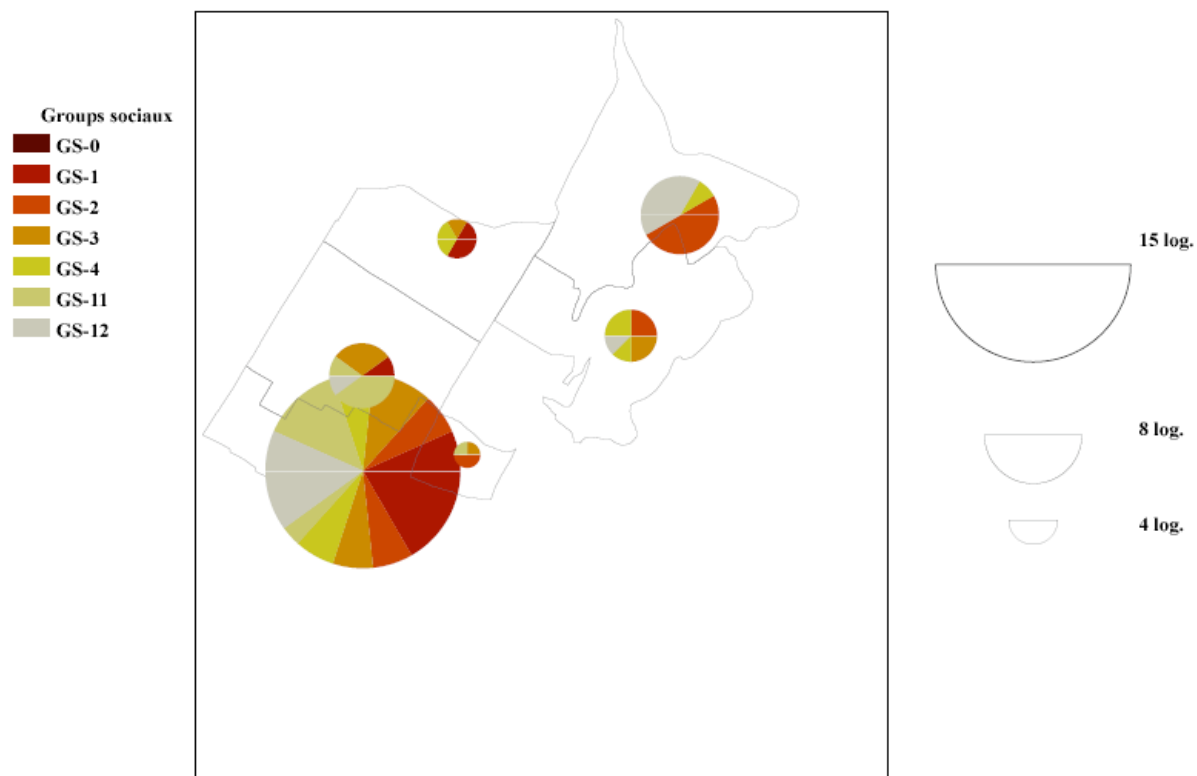


Figure 101. Carte de validation pour le type d'habitat TH-54.

**Validation de l'occupation simulee du type d'habitat TH-55  
pour l'annee 1993**

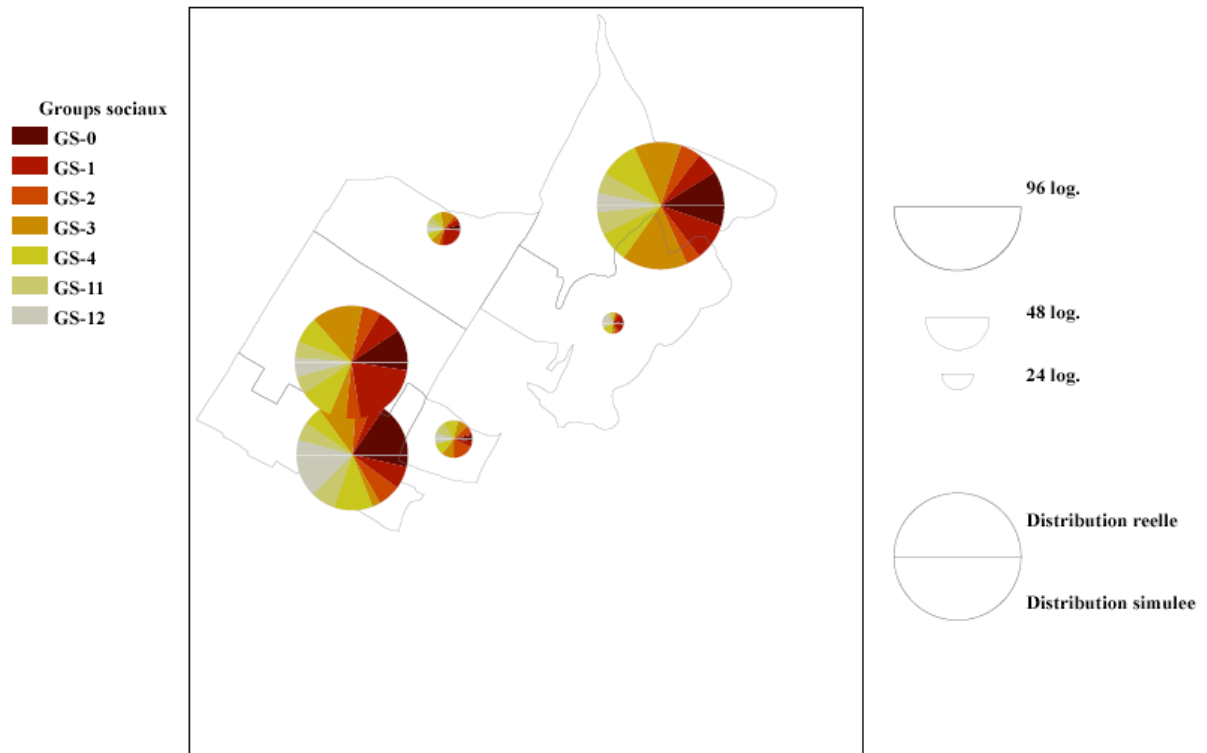


Figure 102. Carte de validation pour le type d'habitat TH-55.

**Validation de l'occupation simulee du type d'habitat TH-56  
pour l'annee 1993**

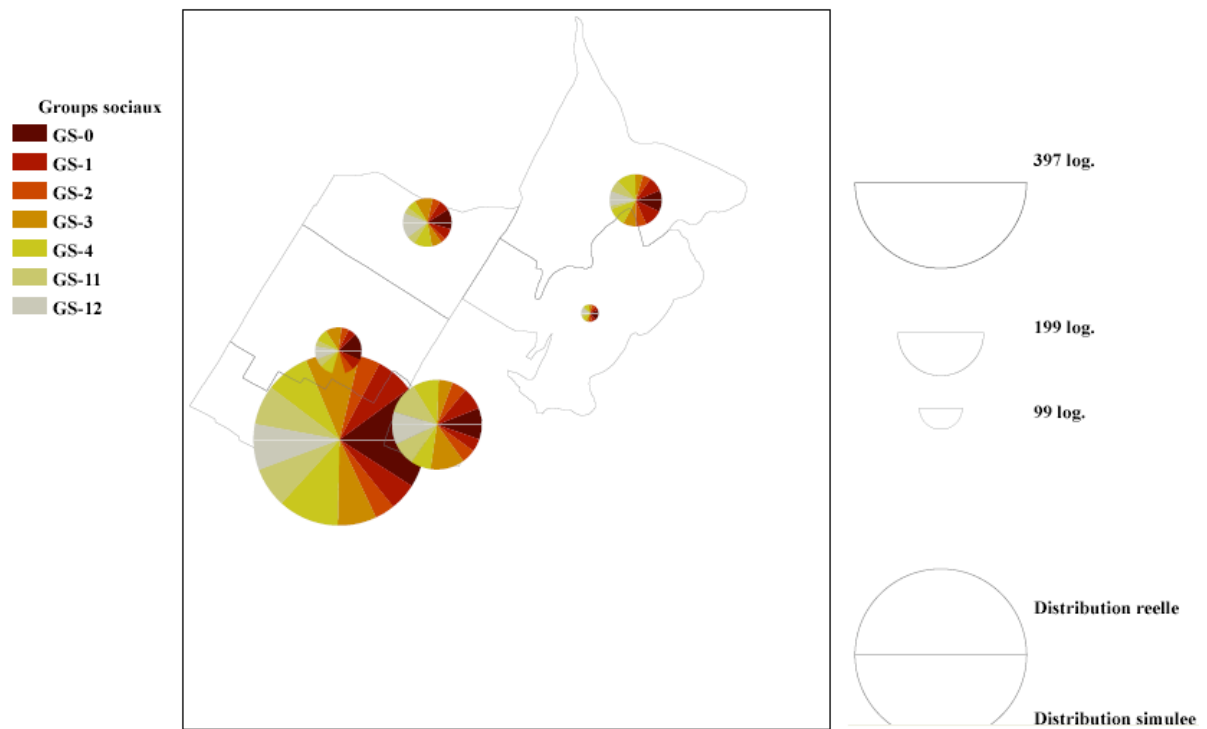


Figure 103. Carte de validation pour le type d'habitat TH-56.

**Validation de l'occupation simulee du type d'habitat TH-57  
pour l'annee 1993**

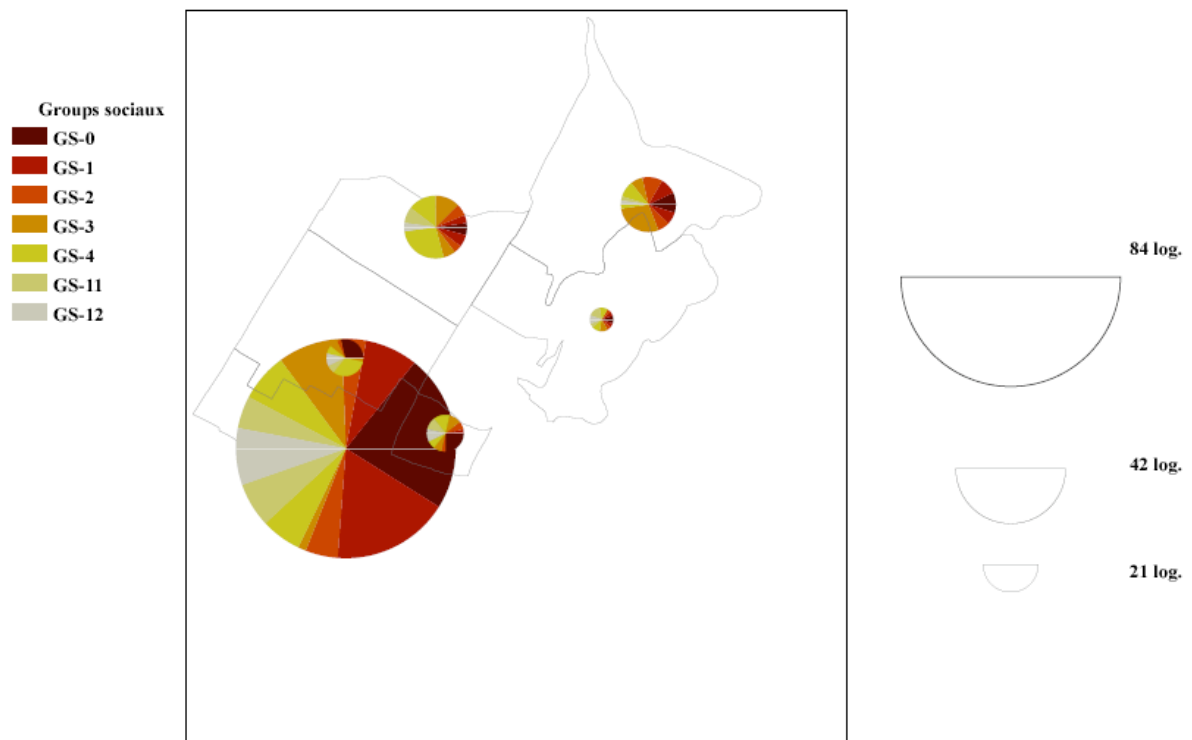


Figure 104. Carte de validation pour le type d'habitat TH-57.

**Validation de l'occupation simulee du type d'habitat TH-58  
pour l'annee 1993**

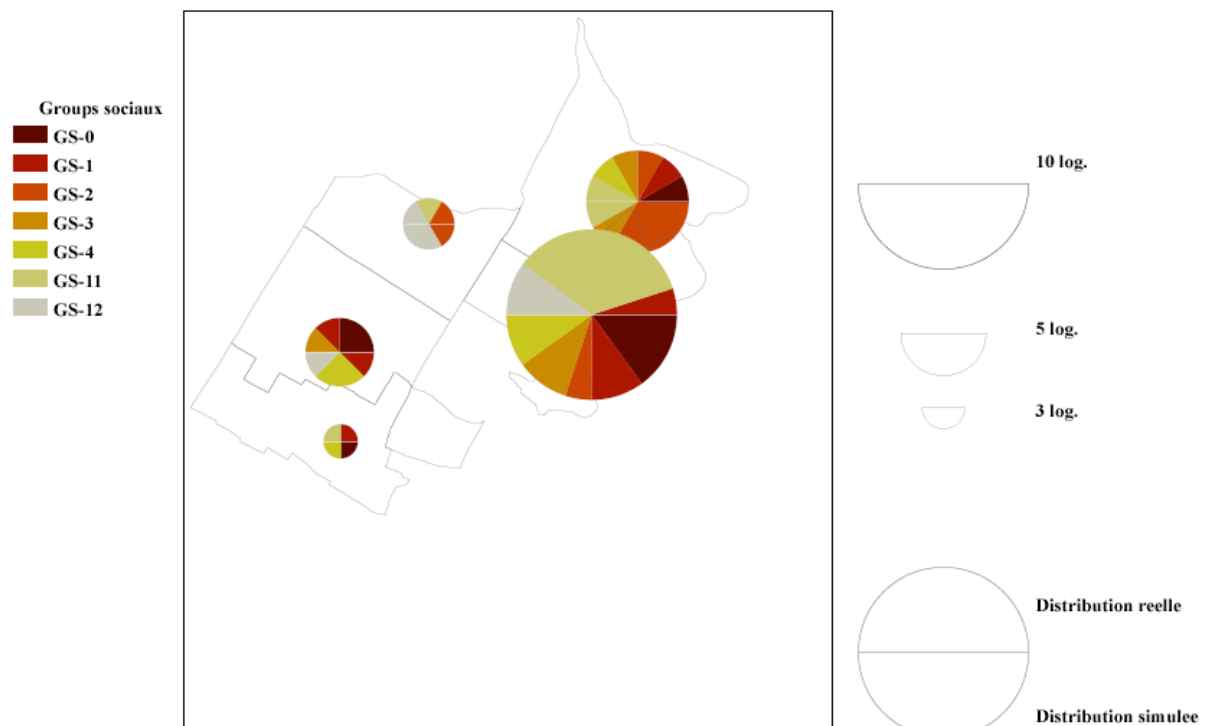


Figure 105. Carte de validation pour le type d'habitat TH-58.



## **Annexe VI. Feuilles XSL pour la transformation des résultats d'une simulation**

Dans cette annexe, nous présentons les principales feuilles de style XSL utilisés dans la transformation des résultats.

### **A6.1. Production de cartes**

Les transformations nécessaires pour produire des cartes de résultats sont programmées sous forme de *templates* de XSL dans la feuille de style *SVGMaps.xsl*. Deux types de cartes sont produites : les cartes de distribution spatiale et les cartes de validation de la distribution de la population simulée par rapport à la population réelle. Pour construire les deux types de cartes, les entrées nécessaires sont :

- les fichiers *socialGroups.xml* et *habitats.xml* (cf. section 10.1.) pour l'itération courante. Pour le cas des cartes de validation, il faut aussi les fichiers *socialGroups.xml* et *habitats.xml* pour l'année de validation (distribution réelle) ;
- le fichier qui contient la description SVG des secteurs urbains considérés (cf. section 10.2.1.) ;
- le fichier *groupsAliases.xml* qui contient la mise en relation des groupes sociaux et types d'habitats ;
- et le repertoire de sortie.

Les cartes de distribution spatiale sont aussi de deux types complémentaires : les cartes de distribution spatiale des groupes sociaux et les cartes d'occupation des types d'habitats. Dans le premier type, on considère à chaque fois la distribution d'un groupe social discriminée par types d'habitat sur l'ensemble des secteurs urbains considérés. Dans le deuxième type, on considère à chaque fois l'occupation de chaque type d'habitat discriminée par groupe social sur l'ensemble des secteurs urbains considérés.

### A6.1.1. Cartes de distribution

#### Template principale

La template *writeMapForGroup* (voir encadré ci-dessous) produit une carte par groupe social et par type d'habitat dans les fichiers *socialGroups.xml* et *habitats.xml*. Les cartes produites sont stockées dans le répertoire de sortie. Le nom de chaque fichier est composé de l'id (attribut *id* dans l'élément *SocialGroup* ou dans l'élément *Habitat*, cf sections 10.1.1. et 10.1.2.) du groupe social ou type d'habitat suivie de l'extension « .svg ».

```
<xsl:template name="writeMapForGroup">
  <xsl:param name="mapNodes"/> <!-- ces nœuds contiennent la représentation sous format svg des secteurs urbains-->
  <xsl:param name="fileName"/> <!-- nom du fichier de sortie -->
  <xsl:param name="isHousing"/> <!-- 0 quand il s'agit des ménages, 1 quand il s'agit des logements -->
  <xsl:param name="groupNode"/> <!-- ce nœud est un nœud SocialGroup ou Habitat selon le cas et contient la description du groupe social ou du type d'habitat représenté-->
  <xsl:param name="groupLabel"/> <!-- alias du groupe social ou type d'habitats obtenu à partir du fichier groupsAliases.xml de la simulation -->
  <xsl:param name="iteration"/> <!-- itération courante -->
  <xsl:param name="labelNodes"/> <!-- noeuds qui contiennent les aliases des groupes sociaux ou des types d'habitats. Si cet carte est une carte d'une groupe social, labelNodes contient les aliases des types d'habitats autrement labelNodes contient les aliases des groupes sociaux. Ils sont utilisés notamment pour construire la legende de la carte et les camemberts décrivant la distribution spatiale-->
  <xsl:call-template name="findLimitsForMap"> <!-- on détermine d'abord la taille de la carte à partir des représentations SVG des secteurs urbains à dessiner-->
    <xsl:with-param name="pathNodes" select="$mapNodes/g"/>
  </xsl:call-template>
  <xsl:result-document href="{ $fileName }"> <!-- On écrit le fichier de sortie -->
  <svg zoomAndPan="disable">
    <xsl:variable name="initX">
      <xsl:choose>
        <xsl:when test="$isHousing = 1">
          <xsl:value-of select="50 + ((round(count($groupsAliases//SocialGroups/SocialGroup) div $maxBricks)+1)*$deltaXBricks)"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select="50 + ((round(count($groupsAliases//Habitats/Habitat) div $maxBricks)+1)*$deltaXBricks)"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
  </svg>
</xsl:template>
```

```

</xsl:variable>
<rect id="mapBorder" x="{ $initX }" y="115.5" width="{ $mapViewPortWidth }"
height="{ $mapViewPortHeight }" style="fill:none;stroke:black;stroke-width:1;pointer-
events:none;shape-rendering:optimizeSpeed" />
<xsl:variable name="factor" select="4221-11062*(number($maxYMap) - number($minYMap))"/>
<xsl:call-template name="writeTitle"> <!-- on écrit le titre -->
    <xsl:with-param name="isHousing" select="$isHousing"/>
    <xsl:with-param name="groupName" select="$groupName/@id"/>
    <xsl:with-param name="iteration" select="$iteration"/>
</xsl:call-template>
<xsl:variable name="maxRadius" select="sum($mapNodes/g/@rmin) div count($mapNodes/g)"/>
<xsl:variable name="maxValue">
    <xsl:for-each select="$groupName/UrbanSectorDistribution/@totalNumber">
        <xsl:sort select="." order="descending" data-type="number"/>
        <xsl:if test="position()=1">
            <xsl:value-of select="."/>
        </xsl:if>
    </xsl:for-each>
</xsl:variable>
<xsl:variable name="higherRadius" select="($maxValue*$maxRadius div
$groupName/@totalNumber)*471 div ($minYMap - $minXMap)"/>
<saxon:assign name="validationRadii" select="0"/>
<saxon:assign name="validationLegendRadii" select="0"/>
<!-- Writes the map -->
<svg x="{ $initX + 5 }" y="60" height="{ $factor }" id="map" preserveAspectRatio="xMinYMin"
viewBox="{ $minXMap } { $minYMap } { $maxXMap } { $maxYMap }" width="{ $factor }">
    <xsl:for-each select="$mapNodes/g">
        <g sector="{ @censusSector }">
            <path d="{ ./path/@d }" style="fill:none;stroke-
width:0.00002204;stroke:rgb(110,110,110)"/>
            <xsl:variable name="censusSector" select="@censusSector"/>
            <xsl:variable name="urbanSectorNode"
select="$groupName/UrbanSectorDistribution[ starts-with(substring-after(@localisation,
'S_URBAN=),$censusSector) ]"/>
            <xsl:variable name="radius"
select="$amplification*$urbanSectorNode/@totalNumber*$maxRadius div
$groupName/@totalNumber"/>
            <xsl:if test="number($radius) &gt; $validationRadii">
                <saxon:assign name="validationRadii" select="$radius"/>
                <saxon:assign name="validationLegendRadii"

```

```

select="$urbanSectorNode/@totalNumber"/>
    </xsl:if>
    <xsl:call-template name="writePieForUrbanSector"> <!-- on écrit le
camembert de la distribution courante -->
        <xsl:with-param name="urbanSectorNode"
select="$urbanSectorNode"/>
        <xsl:with-param name="xCenter" select="@xcenter"/>
        <xsl:with-param name="yCenter" select="@ycenter"/>
        <xsl:with-param name="radius" select="$radius"/>
        <xsl:with-param name="labelNodes" select="$labelNodes"/>
    </xsl:call-template>
    </g>
</xsl:for-each>
</svg>
<!--On écrit la légende de la carte-->
<xsl:call-template name="writeLegend">
    <xsl:with-param name="radii" select="$validationRadii"/>
    <xsl:with-param name="startValue" select="$validationLegendRadii"/>
    <xsl:with-param name="isHousing" select="$isHousing"/>
    <xsl:with-param name="xMinAbsolute" select="$initX +
1.5*$mapViewPortWidth"/>
    <xsl:with-param name="yMinAbsolute" select="125"/>
    <xsl:with-param name="heightAbsolute" select="$mapViewPortWidth"/>
    <xsl:with-param name="widthAbsolute" select="$mapViewPortHeight"/>
    <xsl:with-param name="xMinBox" select="$minXMap"/>
    <xsl:with-param name="yMinBox" select="$minYMap"/>
    <xsl:with-param name="xMaxBox" select="$maxXMap"/>
    <xsl:with-param name="yMaxBox" select="$maxYMap"/>
    <xsl:with-param name="heightBox" select="$factor"/>
    <xsl:with-param name="widthBox" select="$factor"/>
    <xsl:with-param name="labelNodes" select="$labelNodes"/>
</xsl:call-template>
</svg>
</xsl:result-document>
</xsl:template>

```

Encadré 27. Template xsl de création des cartes de distribution spatiale. Ce template fait partie de la feuille de style xsl SVGMap.xsl

### *Templates secondaires*

Les deux templates secondaires les plus importantes sont la template *writePieForUrbanSector* et la template *writeLegend*. La template *writePieForUrbanSector* (voir ci-dessous) crée le camembert pour la distribution du groupe social ou du type d'habitat considéré dans un secteur urbain en particulier. Autrement dit, il écrit sous forme de camembert le contenu d'un nœud *UrbanSectorDistribution*. La template *writeLegend* crée la légende de la carte.

```

<xsl:template name="writePieForUrbanSector">
  <xsl:param name="urbanSectorNode"/> <!--UrbanSectorDistribution à être dessiné-->
  <xsl:param name="xCenter" select="0"/><!--coordonnée x du centre du camembert -->
  <xsl:param name="yCenter" select="0"/><!--coordonnée y du centre du camembert -->
  <xsl:param name="radius"/><!--rayon du camembert -->
  <xsl:param name="labelNodes"/>
  <saxon:assign name="lastAngle" select="0"/>
  <xsl:for-each select="$labelNodes">
    <xsl:sort select="substring-after(@id, '-')" order="ascending" data-type="number"/>
    <xsl:variable name="label" select="@id"/>
    <xsl:variable name="groupName" select="@clusterId"/>
    <xsl:variable name="occupationNode" select="$urbanSectorNode/Occupation[starts-with(@id
, $groupName)]]"/>
    <xsl:if test="$occupationNode">
      <xsl:variable name="color">
        <xsl:call-template name="getColorOnScale">
          <xsl:with-param name="colorNumber" select="position()"/>
          <xsl:with-param name="numberColors"
select="count($labelNodes)"/>
        </xsl:call-template>
      </xsl:variable>
      <xsl:call-template name="writePieSlide">
        <xsl:with-param name="fromAngle" select="$lastAngle"/>
        <xsl:with-param name="value" select="$occupationNode/@number"/>
        <xsl:with-param name="totalValue"
select="$urbanSectorNode/@totalNumber"/>
        <xsl:with-param name="color" select="$color"/>
        <xsl:with-param name="radius" select="$radius"/>
        <xsl:with-param name="xCenter" select="$xCenter"/>
        <xsl:with-param name="yCenter" select="$yCenter"/>
      </xsl:call-template>
    </xsl:if>
  </xsl:for-each>
</xsl:template>

```

Encadré 28. Template xsl de création d'un camembert pour représenter un nœud UrbanSectorDistribution. Ce template fait partie de la feuille de style xsl SVGMap.xsl

```
<!-- This template creates the legend for maps. The legend is composed of bricks on the map's left side and
circles representing size of map's circles plotted on the map's righth side
    it uses the buildsLegendBricks to create bricks and both : writeCircleTextLegends and writeCircles to create
the righthside circles
    @param radii is the max radii in the map
    @param startValue is the value represented by the $radii circle
    @param isHousing 1 if the map is built to represent a housing group's distribution 0 if the map represents a
household group's distribution
    @param xMinAbsolute is the left-upper corner x coordinate of the viewport where circles are plot, it is given
in absolute coordinates
    @param yMinAbsolute is the left-upper corner y coordinate of the viewport where circles are plot, it is given
in absolute coordinates
    @param heighthAbsolute heighth of the viewport in the global coordinates system
    @param widthAbsolute width of the viewport in the global coordinates system
    @param xMinBox is the left-upper corner x coordinate of the viewport where circles are plot, it is given in
viewport coordinates
    @param yMinBox is the left-upper corner y coordinate of the viewport where circles are plot, it is given in
viewport coordinates
    @param xMaxBox is the righth-bottom corner x coordinate of the viewport where circles are plot, it is given in
viewport coordinates
    @param yMaxBox is the righth-bottom corner y coordinate of the viewport where circles are plot, it is given in
viewport coordinates
    @param heighthBox heighth of the viewport in the viewport's coordinates system
    @param widthBox width of the viewport in the viewport's coordinates system
-->
<xsl:template name="writeLegend">
    <xsl:param name="radii" />
    <xsl:param name="startValue" />
    <xsl:param name="isHousing"/>
    <xsl:param name="xMinAbsolute"/>
    <xsl:param name="yMinAbsolute"/>
    <xsl:param name="heightAbsolute"/>
    <xsl:param name="widthAbsolute"/>
    <xsl:param name="xMinBox"/>
    <xsl:param name="yMinBox"/>
    <xsl:param name="xMaxBox"/>
    <xsl:param name="yMaxBox"/>
    <xsl:param name="heightBox"/>
```

```

<xsl:param name="widthBox"/>
<xsl:param name="labelNodes"/>
<!--Builds the bricks -->
<xsl:choose>
  <xsl:when test="$isHousing = '0'">
    <xsl:call-template name="buildsLegendBricks">
      <xsl:with-param name="groupLabelsNodes" select="$labelNodes"/>
      <xsl:with-param name="mainLabel" select="'Habitats'"/>
      <xsl:with-param name="circlesLabel" select="'men.'"/>
    </xsl:call-template>
  </xsl:when>
  <xsl:otherwise>
    <xsl:call-template name="buildsLegendBricks">
      <xsl:with-param name="groupLabelsNodes" select="$labelNodes"/>
      <xsl:with-param name="mainLabel" select="'Groups sociaux'"/>
      <xsl:with-param name="circlesLabel" select="'log.'"/>
    </xsl:call-template>
  </xsl:otherwise>
</xsl:choose>
<!-- Creates the demi circles of the legend -->
<svg x="{ $xMinAbsolute }" y="{ $yMinAbsolute }" height="{ $heightBox }" id="map"
preserveAspectRatio="xMinYMin" viewBox="{ $xMinBox } { $yMinBox } { $xMaxBox } { $yMaxBox }"
width="{ $widthBox }">
  <g>
    <xsl:call-template name="writeCircles">
      <xsl:with-param name="startX" select="$xMinBox + 1.5*$radii"/>
      <xsl:with-param name="startY" select="$yMinBox + 1.5*$radii"/>
      <xsl:with-param name="radii" select="$radii"/>
    </xsl:call-template>
  </g>
</svg>
<!-- Creates the text of the legend text can not bee plot into the viewBox so, we have to remap it outside
the viewBox -->
<xsl:call-template name="writeCircleTextLegends">
  <xsl:with-param name="radii" select="$radii"/>
  <xsl:with-param name="startValue" select="$startValue"/>
  <xsl:with-param name="isHousing" select="$isHousing"/>
  <xsl:with-param name="xMinAbsolute" select="$xMinAbsolute"/>
  <xsl:with-param name="yMinAbsolute" select="$yMinAbsolute"/>
  <xsl:with-param name="heightAbsolute" select="$heightAbsolute"/>

```

```

        <xsl:with-param name="widthAbsolute" select="$widthAbsolute"/>
        <xsl:with-param name="xMinBox" select="$xMinBox"/>
        <xsl:with-param name="yMinBox" select="$yMinBox"/>
        <xsl:with-param name="xMaxBox" select="$xMaxBox"/>
        <xsl:with-param name="yMaxBox" select="$yMaxBox"/>
        <xsl:with-param name="plotDistributionText" select="0"/>
    </xsl:call-template>
</xsl:template>

```

Encadré 29. Template xsl de création de la légende des cartes de distribution spatiale. Ce template fait partie de la feuille de style xsl SVGMap.xsl

### A6.1.2. Cartes de validation

#### *Template principale*

Les cartes de validation de leur côté permettent de comparer les matrices de distribution de la population simulée de la dernière itération et de distribution réelle de l'année finale. Elles sont produites par le template *writeValidationMapForGroup* présenté dans l'encadré ci-dessous :

```

<xsl:template name="writeValidationMapForGroup">
    <xsl:param name="mapNodes"/>
    <xsl:param name="validationMapNodes"/>
    <xsl:param name="fileName"/>
    <xsl:param name="isHousing"/>
    <xsl:param name="groupNode"/>
    <xsl:param name="groupValidationNode"/>
    <xsl:param name="groupLabel"/>
    <xsl:param name="iteration"/>
    <xsl:param name="labelNodes"/>
    <xsl:call-template name="findLimitsForMap">
        <xsl:with-param name="pathNodes" select="$mapNodes/g"/>
    </xsl:call-template>
    <xsl:result-document href="{ $fileName }">
        <svg zoomAndPan="disable">
            <xsl:variable name="initX">
                <xsl:choose>
                    <xsl:when test="$isHousing = 1">
                        <xsl:value-of select="50 +
((round(count($groupsAliases//SocialGroups/Group) div $maxBricks)+1)*$deltaXBricks)"/>
                    </xsl:when>
                </xsl:choose>
            </xsl:variable>

```



```

<xsl:otherwise>
    <xsl:value-of select="50 +
((round(count($groupsAliases//Habitats/Group) div $maxBricks)+1)*$deltaXBricks)"/>
</xsl:otherwise>
</xsl:choose>
</xsl:variable>
<xsl:variable name="factor" select="4221-11062*(number($maxYMap) - number($minYMap))"/>
<xsl:call-template name="writeValidationTitle">
    <xsl:with-param name="isHousing" select="$isHousing"/>
    <xsl:with-param name="groupName" select="$groupName/@id"/>
    <xsl:with-param name="iteration" select="$iteration"/>
</xsl:call-template>
<xsl:variable name="maxRadius" select="sum($mapNodes/g/@rmin) div
count($mapNodes/g)"/>
<xsl:variable name="maxRadiusValidation">
    <xsl:choose>
        <xsl:when test="$validationMapNodes"><xsl:value-of
select="sum($validationMapNodes/g/@rmin) div count($validationMapNodes/g)"/></xsl:when>
        <xsl:otherwise>0</xsl:otherwise>
    </xsl:choose>
</xsl:variable>
<xsl:variable name="maxValue">
    <xsl:for-each
select="$groupName/UrbanSectorDistribution/@totalNumber">
        <xsl:sort select="." order="descending" data-
type="number"/>
        <xsl:if test="position() =1">
            <xsl:value-of select="."/>
        </xsl:if>
    </xsl:for-each>
</xsl:variable>
<xsl:variable name="maxValueValidation">
    <xsl:choose>
        <xsl:when test="$groupNameValidation">
            <xsl:for-each
select="$groupNameValidation/UrbanSectorDistribution/@totalNumber">
                <xsl:sort select="." order="descending"
data-type="number"/>
                <xsl:if test="position() =1">
                    <xsl:value-of select="."/>

```

```

        </xsl:if>
        </xsl:for-each>
        </xsl:when>
        <xsl:otherwise>0</xsl:otherwise>
    </xsl:choose>
</xsl:variable>
    <xsl:variable name="higherRadius" select="($maxValue*$maxRadius div
$groupNode/@totalNumber)*471 div ($minYMap - $minXMap)"/>
    <xsl:variable name="higherValidaitonRadius">
        <xsl:choose>
            <xsl:when test="$validationMapNodes">
                <xsl:value-of
select="($maxValueValidation*$maxRadiusValidation div $groupValidationNode/@totalNumber)*471 div
($minYMap - $minXMap)"/>
            </xsl:when>
            <xsl:otherwise>0</xsl:otherwise>
        </xsl:choose>
</xsl:variable>
    <saxon:assign name="validationRadii" select="0"/>
    <saxon:assign name="validationLegendRadii" select="0"/>
    <!-- Writes the map -->
    <svg x="{ $initX + 5}" y="60" height="{ $factor}" id="map"
preserveAspectRatio="xMinYMin" viewBox="{ $minXMap} { $minYMap} { $maxXMap} { $maxYMap}"
width="{ $factor}">
        <xsl:for-each select="$mapNodes/g">
            <g sector="{ @censusSector}">
                <path d="{ ./path/@d}" style="fill:none;stroke-
width:0.00002204;stroke:rgb(110,110,110)"/>
                <xsl:variable name="censusSector"
select="@censusSector"/>
                <xsl:variable name="urbanSectorNode"
select="$groupNode/UrbanSectorDistribution[ starts-with(substring-after(@localisation,
'S_URBAN=),$censusSector) ]"/>
                <xsl:variable name="radius"
select="$amplification*$urbanSectorNode/@totalNumber*$maxRadius div $groupNode/@totalNumber"/>
                <xsl:if test="number($radius) >
$validationRadii">
                    <saxon:assign name="validationRadii"
select="$radius"/>
                    <saxon:assign

```

```

name="validationLegendRadii" select="$urbanSectorNode/@totalNumber"/>
    </xsl:if>

    <xsl:call-template
name="writeSemiPieForUrbanSector">
        <xsl:with-param
name="urbanSectorNode" select="$urbanSectorNode"/>
        <xsl:with-param name="xCenter"
select="@xcenter"/>
        <xsl:with-param name="yCenter"
select="@ycenter"/>
        <xsl:with-param name="radius"
select="$radius"/>
        <xsl:with-param name="angleSign"
select="1"/>
        <xsl:with-param name="labelNodes"
select="$labelNodes"/>
        <xsl:with-param name="validation"
select="0"/>

    </xsl:call-template>
    <xsl:if test="$groupValidationNode">
        <xsl:variable
name="urbanSectorValidationNode" select="$groupValidationNode/UrbanSectorDistribution[
starts-with(substring-after(@localisation, 'S_URBAN='),$scensusSector)]"/>
        <xsl:variable name="radiusValidation"
select="$amplification*$urbanSectorValidationNode/@totalNumber*$maxRadiusValidation div
$groupValidationNode/@totalNumber"/>
        <xsl:if test="number($radiusValidation)
&gt; $validationRadii">
            <saxon:assign
name="validationRadii" select="$radiusValidation"/>
            <saxon:assign
name="validationLegendRadii" select="$urbanSectorValidationNode/@totalNumber"/>
        </xsl:if>
    <xsl:call-template
name="writeSemiPieForUrbanSector">
        <xsl:with-param
name="urbanSectorNode" select="$urbanSectorValidationNode"/>
        <xsl:with-param

```

```

name="xCenter" select="@xcenter"/>
<xsl:with-param
name="yCenter" select="@ycenter"/>
<xsl:with-param name="radius"
select="$radiusValidation"/>
<xsl:with-param
name="angleSign" select="-1"/>
<xsl:with-param
name="labelNodes" select="$labelNodes"/>
<xsl:with-param
name="validation" select="1"/>
</xsl:call-template>
</xsl:if>
</g>
</xsl:for-each>
<xsl:if test="$validationMapNodes">
<xsl:for-each select="$validationMapNodes/g">
<xsl:variable name="censusSector"
select="@censusSector"/>
<xsl:if test="not($mapNodes/g[starts-with(@censusSector,
$censusSector)])">
<g sector="{@censusSector}">
<path d="{./path/@d}"
style="fill:none;stroke-width:0.00002204;stroke:rgb(110,110,110)"/>
<xsl:variable
name="urbanSectorValidationNode" select="$groupValidationNode/UrbanSectorDistribution[ starts-
with(substring-after(@localisation, 'S_URBAN='),$censusSector) ]"/>
<xsl:variable name="radiusValidation"
select="$amplification*$urbanSectorValidationNode/@totalNumber*$maxRadiusValidation div
$groupValidationNode/@totalNumber"/>
<xsl:if test="number($radiusValidation)
> $validationRadii">
<saxon:assign
name="validationRadii" select="$radiusValidation"/>
<saxon:assign
name="validationLegendRadii" select="$urbanSectorValidationNode/@totalNumber"/>
</xsl:if>
<xsl:call-template
name="writeSemiPieForUrbanSector">

```

```

name="urbanSectorNode" select="$urbanSectorValidationNode"/>
name="xCenter" select="@xcenter"/>
name="yCenter" select="@ycenter"/>
select="$radiusValidation"/>
name="angleSign" select="-1"/>
name="labelNodes" select="$labelNodes"/>
name="validation" select="1"/>
</xsl:call-template>
</g>
</xsl:if>
</xsl:for-each>
</xsl:if>
</svg>
<!--Writes the legend -->
<xsl:call-template name="writeValidationLegend">
  <xsl:with-param name="radii" select="$validationRadii"/>
  <xsl:with-param name="startValue"
select="$validationLegendRadii"/>
  <xsl:with-param name="isHousing" select="$isHousing"/>
  <xsl:with-param name="xMinAbsolute" select="$initX +
1.5*$mapViewPortWidth"/>
  <xsl:with-param name="yMinAbsolute" select="125"/>
  <xsl:with-param name="heightAbsolute"
select="$mapViewPortWidth"/>
  <xsl:with-param name="widthAbsolute"
select="$mapViewPortHeigth"/>
  <xsl:with-param name="xMinBox" select="$minXMap"/>
  <xsl:with-param name="yMinBox" select="$minYMap"/>
  <xsl:with-param name="xMaxBox" select="$maxXMap"/>
  <xsl:with-param name="yMaxBox" select="$maxYMap"/>
  <xsl:with-param name="heightBox" select="$factor"/>
  <xsl:with-param name="widthBox" select="$factor"/>
  <xsl:with-param name="labelNodes" select="$labelNodes"/>

```

```

        </xsl:call-template>
    </svg>
</xsl:result-document>
</xsl:template>

```

Encadré 30. Template xsl de création des cartes de valiation de la distribution spatiale. Ce template fait partie de la feuille de style xsl *SVGMap.xsl*

### Templates secondaires

```

<xsl:template name="writeSemiPieForUrbanSector">
    <xsl:param name="urbanSectorNode"/>
    <xsl:param name="xCenter" select="0"/>
    <xsl:param name="yCenter" select="0"/>
    <xsl:param name="radius"/>
    <xsl:param name="angleSign" select="1"/>
    <xsl:param name="labelNodes"/>
    <xsl:param name="validation" select="0"/>
    <saxon:assign name="lastAngle" select="0"/>
    <xsl:for-each select="$labelNodes">
        <xsl:sort select="substring-after(@id, '-')" order="ascending" data-type="number"/>
        <xsl:variable name="label" select="@id"/>
        <xsl:variable name="groupName">
            <xsl:choose>
                <xsl:when test="$validation = 0">
                    <xsl:value-of select="@clusterId"/>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:value-of select="@validationId"/>
                </xsl:otherwise>
            </xsl:choose>
        </xsl:variable>
        <xsl:variable name="occupationNode" select="$urbanSectorNode/Occupation[starts-with(@id, $groupName)]"/>
        <xsl:if test="$occupationNode">
            <xsl:variable name="color">
                <xsl:call-template name="getColorOnScale">
                    <xsl:with-param name="colorNumber" select="position()"/>
                    <xsl:with-param name="numberColors"
select="count($labelNodes)"/>
                </xsl:call-template>
            </xsl:variable>

```

```

        <xsl:call-template name="writePieSlide">
            <xsl:with-param name="fromAngle" select="$lastAngle"/>
            <xsl:with-param name="value" select="$occupationNode/@number div
2"/><!--It does just a semi circle-->
            <xsl:with-param name="totalValue"
select="$urbanSectorNode/@totalNumber"/>
            <xsl:with-param name="color" select="$color"/>
            <xsl:with-param name="radius" select="$radius"/>
            <xsl:with-param name="xCenter" select="$xCenter"/>
            <xsl:with-param name="yCenter" select="$yCenter"/>
            <xsl:with-param name="angleSign" select="$angleSign"/>
        </xsl:call-template>
    </xsl:if>
</xsl:for-each>
    <path d="M{$xCenter - $radius}, {$yCenter}L{$xCenter + $radius}, {$yCenter}"
style="fill:none;stroke-width:{$radius div 1000};stroke:rgb(255,255,255)"/>
</xsl:template>

```

Encadré 31. Template xsl de création de semi-camemberts dans des cartes de validation de la distribution spatiale. Ce template fait partie de la feuille de style xsl SVGMap.xsl

```

<!-- This template creates the legend for validation maps. The legend is composed of bricks on the map's left
side and demi circles representing size of map's demi-circles plotted on the map's righth side
    it uses the buildsLegendBricks to create bricks and both : writeCircleTextLegends and writeDemiCircles to
create the righthside demi-circles
    @param radii is the max radii in the map
    @param startValue is the value represented by the $radii circle
    @param isHousing 1 if the map is built to represent a housing group's distribution 0 if the map represents a
household group's distribution
    @param xMinAbsolute is the left-upper corner x coordinate of the viewport where demi-circles are plot, it is
given in absolute coordinates
    @param yMinAbsolute is the left-upper corner y coordinate of the viewport where demi-circles are plot, it is
given in absolute coordinates
    @param heigthAbsolute heigth of the viewport in the global coordinates system
    @param widthAbsolute width of the viewport in the global coordinates system
    @param xMinBox is the left-upper corner x coordinate of the viewport where demi-circles are plot, it is given
in viewport coordinates
    @param yMinBox is the left-upper corner y coordinate of the viewport where demi-circles are plot, it is given
in viewport coordinates
    @param xMaxBox is the righth-bottom corner x coordinate of the viewport where demi-circles are plot, it is
given in viewport coordinates

```

@param yMaxBox is the right-bottom corner y coordinate of the viewport where demi-circles are plot, it is given in viewport coordinates

@param heigthBox heigth of the viewport in the viewport's coordinates system

@param widthBox width of the viewport in the viewport's coordinates system

-->

```
<xsl:template name="writeValidationLegend">
  <xsl:param name="radii" />
  <xsl:param name="startValue" />
  <xsl:param name="isHousing"/>
  <xsl:param name="xMinAbsolute"/>
  <xsl:param name="yMinAbsolute"/>
  <xsl:param name="heigthAbsolute"/>
  <xsl:param name="widthAbsolute"/>
  <xsl:param name="xMinBox"/>
  <xsl:param name="yMinBox"/>
  <xsl:param name="xMaxBox"/>
  <xsl:param name="yMaxBox"/>
  <xsl:param name="heigthBox"/>
  <xsl:param name="widthBox"/>
  <xsl:param name="labelNodes"/>
  <!--Builds the bricks -->
  <xsl:choose>
    <xsl:when test="$isHousing = '0'">
      <xsl:call-template name="buildsLegendBricks">
        <xsl:with-param name="groupLabelsNodes" select="$labelNodes"/>
        <xsl:with-param name="mainLabel" select="'Habitats'"/>
        <xsl:with-param name="circlesLabel" select="'men.'"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name="buildsLegendBricks">
        <xsl:with-param name="groupLabelsNodes" select="$labelNodes"/>
        <xsl:with-param name="mainLabel" select="'Groups sociaux'"/>
        <xsl:with-param name="circlesLabel" select="'log.'"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
  <!-- Creates the demi circles of the legend -->
  <svg x="{ $xMinAbsolute }" y="{ $yMinAbsolute }" height="{ $heigthBox }" id="map"
  preserveAspectRatio="xMinYMin" viewBox="{ $xMinBox } { $yMinBox } { $xMaxBox } { $yMaxBox }"
```



```

width="{ $widthBox }">
    <g>
        <xsl:call-template name="writeDemiCircles">
            <xsl:with-param name="startX" select="$xMinBox + 1.5*$radii"/>
            <xsl:with-param name="startY" select="$yMinBox + 1.5*$radii"/>
            <xsl:with-param name="radii" select="$radii"/>
        </xsl:call-template>
        <xsl:variable name="xCenter" select="$xMinBox + 8*$radii"/>
        <xsl:variable name="yCenter" select="$yMinBox + 1.5*$radii"/>
        <circle cx="{ $xCenter }" cy="{ $yCenter }" r="{ $radii }" style="fill:none;stroke-
width: {0.01*$radii};stroke:rgb(110,110,110)"/>
        <path d="M{ $xCenter - $radii }, { $yCenter } L{ $xCenter + $radii }, { $yCenter }"
style="fill:none;stroke-width: {0.01*$radii};stroke:rgb(110,110,110)"/>
    </g>
</svg>
<!-- Creates the text of the legend text can not be plot into the viewBox so, we have to remap it outside
the viewBox -->
<xsl:call-template name="writeCircleTextLegends">
    <xsl:with-param name="radii" select="$radii"/>
    <xsl:with-param name="startValue" select="$startValue"/>
    <xsl:with-param name="isHousing" select="$isHousing"/>
    <xsl:with-param name="xMinAbsolute" select="$xMinAbsolute"/>
    <xsl:with-param name="yMinAbsolute" select="$yMinAbsolute"/>
    <xsl:with-param name="heightAbsolute" select="$heightAbsolute"/>
    <xsl:with-param name="widthAbsolute" select="$widthAbsolute"/>
    <xsl:with-param name="xMinBox" select="$xMinBox"/>
    <xsl:with-param name="yMinBox" select="$yMinBox"/>
    <xsl:with-param name="xMaxBox" select="$xMaxBox"/>
    <xsl:with-param name="yMaxBox" select="$yMaxBox"/>
</xsl:call-template>
</xsl:template>

```

Encadré 32. Template xsl de création de la légende des cartes de validation de la distribution spatiale. Ce template fait partie de la feuille de style xsl SVGMap.xsl

## A6.2 Diagrammes et tableaux statistiques

Les transformations nécessaires pour produire des tableaux de caractérisation automatique (cf. section 10.2.3.), des diagrammes de valeur test (cf. section 10.2.2.) et des diagrammes historiques (cf. section 10.2.4.) des groupes sociaux et des types d'habitat sont programmés sous forme de templates de XSL dans la feuille de style *Statistics.xsl*. Les templates principales de production des tableaux et diagrammes sont présentées ci-après.

### A6.2.1. Production de diagrammes de description automatique des groupes sociaux et des types d'habitat

Afin de faciliter la construction des diagrammes de description automatique, on effectue un précalcul des différentes statistiques nécessaires à la détermination des valeurs de test de chaque groupe social ou type d'habitats. Ce précalcul est réalisé par la template *buildStatisticsForGroup*, présentée dans l'encadré ci-dessous. Cette template construit des valises qui contiennent les différentes statistiques calculées pour un groupe social ou type d'habitat donné. Les résultats produits par cet template sont stockés dans un fichier temporelle qui est par la suite utilisé pour produire les diagrammes de description automatique et les tableaux récapitulatifs de description automatique.

```
<xsl:template name="buildStatisticsForGroup">
  <xsl:param name="groupId"/>
  <xsl:param name="rootNode"/>
  <xsl:param name="variablesNodes"/>
  <xsl:param name="totalPopulation"/>
  <xsl:variable name="groupAlias">
    <xsl:call-template name="getLabelForGroup">
      <xsl:with-param name="groupName" select="$groupId"/>
      <xsl:with-param name="aliases" select="$groupsAliases"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:variable name="group" select="$rootNode//Group[contains(@id , $groupId)]"/>
  <xsl:variable name="groupPercentages" select="$rootNode//GroupPercentages[contains(@id , $groupId)]"/>
  <xsl:variable name="populationModalities"
select="$rootNode//DistributionOnTotalPopulation/@populationModalities"/>
```

```

<saxon:assign name="indexVariable" select="0"/>
<xsl:variable name="percentage" select="$group/@totalNumber div $totalPopulation"/>
<GroupStatistics id="{ $group/@id}" alias="{ $groupAlias}" nk="{ $group/@totalNumber}"
nkOnN="{ $percentage}">
  <xsl:for-each select="$variablesNodes">
    <xsl:variable name="modalities" select="./Modality"/>
    <Variable name="{ name/@value}">
      <xsl:for-each select="$modalities">
        <xsl:variable name="effectif">
          <xsl:call-template name="displayVectorElement">
            <xsl:with-param name="vector"
select="$group/RealTagVector/@vector"/>
              <xsl:with-param name="index" select="$indexVariable"/>
            </xsl:call-template>
          </xsl:variable>
          <xsl:variable name="modalityGroupPercentage">
            <xsl:call-template name="displayVectorElement">
              <xsl:with-param name="vector"
select="$group/RealTagVector/@histogramme"/>
                <xsl:with-param name="index" select="$indexVariable"/>
            </xsl:call-template>
          </xsl:variable>
          <xsl:variable name="modalityGroupOnPopulationPercentage">
            <xsl:call-template name="displayVectorElement">
              <xsl:with-param name="vector"
select="$groupPercentages/@percentages"/>
                <xsl:with-param name="index" select="$indexVariable"/>
                <xsl:with-param name="separator" select="" ""/>
            </xsl:call-template>
          </xsl:variable>
          <xsl:variable name="totalPopulationEffectif">
            <xsl:call-template name="displayVectorElement">
              <xsl:with-param name="vector"
select="$populationModalities"/>
                <xsl:with-param name="index" select="$indexVariable"/>
                <xsl:with-param name="separator" select="" ""/>
            </xsl:call-template>
          </xsl:variable>
          <xsl:variable name="populationModalityPercentage"
select="$totalPopulationEffectif div $totalPopulation"/>

```

```

        <Modality name="{@value}">
            <General njk="{ $effectif}" nj="{ $totalPopulationEffectif}"
njkOnNk="{ $modalityGroupPercentage div 100 }" njkOnNj="{ $modalityGroupOnPopulationPercentage div
100}" njOnN="{ $populationModalityPercentage}"/>
                <xsl:variable name="ek">
                    <xsl:call-template name="calculateEn">
                        <xsl:with-param name="nk"
select="$group/@totalNumber"/>
                        <xsl:with-param name="nj"
select="$totalPopulationEffectif"/>
                        <xsl:with-param name="n"
select="$totalPopulation"/>
                    </xsl:call-template>
                </xsl:variable>
                <xsl:variable name="s2k">
                    <xsl:call-template name="calculateS2k">
                        <xsl:with-param name="nk"
select="$group/@totalNumber"/>
                        <xsl:with-param name="nj"
select="$totalPopulationEffectif"/>
                        <xsl:with-param name="n"
select="$totalPopulation"/>
                    </xsl:call-template>
                </xsl:variable>
                <xsl:variable name="testValue">
                    <xsl:call-template name="calculateTestValue">
                        <xsl:with-param name="En" select="$ek"/>
                        <xsl:with-param name="s2k" select="$s2k"/>
                        <xsl:with-param name="nj" select="$effectif"/>
                    </xsl:call-template>
                </xsl:variable>
                <xsl:if test="position() = 1">
                    <saxon:assign name="maxTestValue"
select="$testValue"/>
                    <saxon:assign name="maxModality" select="@value"/>
                </xsl:if>
                <xsl:if test="not(position() = 1) and $testValue >
$maxTestValue">
                    <saxon:assign name="maxTestValue"
select="$testValue"/>

```

```

                <saxon:assign name="maxModality" select="@value"/>
            </xsl:if>
            <TestValue ek="{ $ek}" s2k="{ $s2k}" value="{ $testValue}"/>
            <saxon:assign name="indexVariable" select="$indexVariable+1"/>
        </Modality>
    </xsl:for-each>
    <ModalityMaxTestValue name="{ $maxModality}"
testValue="{ $maxTestValue}"/>
    </Variable>
</xsl:for-each>
</GroupStatistics>
</xsl:template>

```

Encadré 33. Template xsl de précalcul des statistiques utilisées dans la détermination des valeurs test des groupes sociaux et des types d'habitats. Ce template fait partie de la feuille de style xsl Statistics.xsl

Les diagrammes de description automatique sont construits par la template *writeTestValueDescriptionHistogramme* (présentée dans l'encadré ci-dessous) à partir de la sortie produite par la template *buildStatisticsForGroup*. La template *writeTestValueDescriptionHistogramme* produit un fichier en format SVG qui contient le résultat des valeurs-test sous forme d'histogramme descendant (les modalités qui ont les valeurs-test les plus forts sont présentées en premier) d'un groupe social ou d'un type d'habitats.

```

<xsl:template name="writeTestValueDescriptionHistogramme">
    <xsl:param name="fileName"/>
    <xsl:param name="statisticsNode"/> <!-- noeud de statistiques produit par la template
buildStatisticsForGroup -->
    <xsl:param name="scale" select="'2'"/>
    <xsl:param name="iterationLabel"/>
    <xsl:call-template name="findHistogrammeSizesForStatisticNode">
        <xsl:with-param name="statisticsNode" select="$statisticsNode"/>
    </xsl:call-template>
    <xsl:result-document href="{ $fileName}"/>
    <svg width="100%" height="100%">
        <g transform="translate(130, {(number($scale)*100)+130}) scale({ $scale}, -
{ $scale})">
            <xsl:call-template name="writeSVGHistogrammeDefs">
            </xsl:call-template>
            <xsl:call-template name="createAxes">
                <xsl:with-param name="width" select="$lastWidth"/>
                <xsl:with-param name="height" select="$lastMax * 1.5"/>

```

```

</xsl:call-template>
<!--Plots the bars-->
<xsl:for-each select="$statisticsNode//ModalityMaxTestValue">
  <xsl:sort data-type="number" select="@testValue"
order="descending"/>
  <xsl:variable name="textOnBar">
    <xsl:call-template name="printRounded">
      <xsl:with-param name="number"
select="@testValue"/>
    </xsl:call-template>
  </xsl:variable>
  <xsl:call-template name="createSVGBar">
    <xsl:with-param name="x" select="(position()-1)*30"/>
    <xsl:with-param name="height" select="@testValue"/>
    <xsl:with-param name="legend" select="concat('./@name,'
: ', @name)"/>
    <xsl:with-param name="textOnBar" select="$textOnBar"/>
  </xsl:call-template>
</xsl:for-each>
<xsl:call-template name="createSVGGraduations">
  <xsl:with-param name="titleY" select="'V.T.'"/>
  <xsl:with-param name="height" select="$lastMax * 1.5"/>
  <xsl:with-param name="mainUnit" select="'10'"/>
</xsl:call-template>
</g>
<g transform="scale({$scale},{$scale})">
  <xsl:variable name="title">
    <xsl:text>Caractérisation automatique du groupe </xsl:text><xsl:value-of
select="$statisticsNode/@alias"/><xsl:text>pour l'année </xsl:text><xsl:value-of select="$iterationLabel"/>
  </xsl:variable>
  <xsl:variable name="sub_title" select="'Selon les valeurs de test'"/>
  <xsl:call-template name="addTitle">
    <xsl:with-param name="text" select="$title"/>
    <xsl:with-param name="x" select="65"/>
    <xsl:with-param name="y" select="15"/>
  </xsl:call-template>
  <xsl:call-template name="addTitle">
    <xsl:with-param name="text" select="$sub_title"/>
    <xsl:with-param name="x" select="65"/>
    <xsl:with-param name="y" select="30"/>
  </xsl:call-template>

```

```

        </xsl:call-template>
    </g>
</svg>
</xsl:result-document>
</xsl:template>

```

Encadré 34. Template xsl de création de diagrammes de description automatique des groupes sociaux et des types d'habitats. Ce template fait partie de la feuille de style xsl Statistics.xsl

## A6.2.2. Production de tableaux récapitulatifs de description automatique des groupes sociaux et des types d'habitat

Les tableaux récapitulatifs de description automatique pour un groupe social ou un type d'habitats sont construits par la template `printStatisticsForGroup` (présentée dans l'encadré ci-dessous). Ces résultats sont présentés sous forme de tableau *html*. Ils sont produits à partir de la sortie produite par la template `buildStatisticsForGroup`.

```

<xsl:template name="printStatisticsForGroup">
    <xsl:param name="statisticNode"/> <!-- noeud de statistiques produit par la template
buildStatisticsForGroup -->
    <xsl:param name="totalPopulation"/>
    <xsl:param name="iterationLabel"/>
    <a name="{ $statisticNode/@id }"><div align="center"> <b>Analyse de la distribution des modalités
pour le group : <xsl:value-of select="$statisticNode/@alias"/>(<xsl:value-of select="$statisticNode/@id"/>)
<br/>Pour l'année <xsl:value-of select="$iterationLabel"/>
</b>
</div></a>
    <div align="center"> <b>Nombre d'effectifs : <xsl:value-of select="$statisticNode/@nk"/> (
        <xsl:call-template name="printPourcentage">
            <xsl:with-param name="number" select="$statisticNode/@nkOnN"/>
        </xsl:call-template>
    )sur une population de <xsl:value-of select="$totalPopulation"/> individus
    </b>
</div>
<div><a href="#indexGroups">Retour à l'index des groups</a></div>
<table>
    <tr bgcolor="#5ebbc7">
        <th rowspan="1" align="center">Variable</th>
        <th rowspan="1" align="center">Modalités</th>
        <th rowspan="1" align="center">Effectifs dans la modalité</th>
        <th rowspan="1" align="center">% modalité dans le group par rapport au group

```

```

Les pourcentages en
<font face="verdana, geneva, arial" size="2" color="#ff0000">
rouge
</font> sont plus grands que le pourcentage de la modalité dans la
population

</th>
<th rowspan="1" align="center">% modalité dans le group par rapport à la population
</th>
<th rowspan="1" align="center">% de la modalité dans la population</th>
<th rowspan="1" align="center">Valeur test</th>

</tr>
<tr bgcolor="#5ebbc7"><th/><th/><th/><th/><th/><th/></tr>
<xsl:for-each select="$StatisticNode/Variable">
  <xsl:variable name="modalities" select="./Modality"/>
  <tr bgcolor="#ffffaa">
    <td rowspan="{count($modalities)+1}" bgcolor="#40838c" align="left">
      <font face="verdana, geneva, arial" size="2" color="#ffffff">
        <xsl:value-of select="@name"/>
      </font>
    </td>
  </tr>
  <xsl:for-each select="$modalities">
    <xsl:sort select="./TestValue/@value" order="descending" data-
type="number"/>
    <xsl:variable name="effectif" select="./General/@njc"/>
    <xsl:variable name="modalityGroupPercentage"
select="./General/@njcOnNk"/>
    <xsl:variable name="modalityGroupOnPopulationPercentage"
select="./General/@njcOnNj"/>
    <xsl:variable name="totalPopulationEffectif" select="./General/@nj"/>
    <xsl:variable name="populationModalityPercentage"
select="./General/@njOnN"/>
    <tr bgcolor="#ffffaa">
      <td>
        <xsl:value-of select="@name"/>
      </td>
      <td>
        <xsl:value-of select="$effectif"/>
      </td>

```



```

        <td>
        <xsl:choose>
            <xsl:when test=" $populationModalityPercentage &lt;
$modalityGroupPercentage ">
                <font face="verdana, geneva, arial" size="2"
color="#ff0000">
                    <xsl:call-template name="printPourcentage">
                        <xsl:with-param name="number"
select="$modalityGroupPercentage"/>
                    </xsl:call-template>
                </font>
            </xsl:when>
            <xsl:otherwise>
                <xsl:call-template name="printPourcentage">
                    <xsl:with-param name="number"
select="$modalityGroupPercentage"/>
                </xsl:call-template>
            </xsl:otherwise>
        </xsl:choose>
    </td>
    <td>
        <xsl:call-template name="printPourcentage">
            <xsl:with-param name="number"
select="$modalityGroupOnPopulationPercentage"/>
        </xsl:call-template>
    </td>
    <td>
        <xsl:call-template name="printPourcentage">
            <xsl:with-param name="number"
select="$populationModalityPercentage"/>
        </xsl:call-template>
    </td>
    <td>
        <xsl:call-template name="printRounded">
            <xsl:with-param name="number"
select="./TestValue/@value"/>
        </xsl:call-template>
    </td>
    <saxon:assign name="indexVariable" select="$indexVariable+1"/>
</tr>

```

```

        </xsl:for-each>
        <tr bgcolor="#5ebbc7"><th/><th/><th/><th/><th/><th/></tr>
    </xsl:for-each>
</table>
</xsl:template>

```

Encadré 35. Template xsl de création de tableaux récapitulatifs de description automatique des groupes sociaux et des types d'habitats. Ce template fait partie de la feuille de style xsl Statistics.xsl

### A6.2.3. Production de diagrammes historiques des groupes sociaux et des types d'habitats

Les templates XSL, nécessaires pour produire les diagrammes d'évolution historique des groupes sociaux et des types d'habitats (cf. section 10.2.3.), sont programmés dans la feuille de style *Evolution.xsl*. La template *createTempHistoricForGroups* produit un fichier temporaire qui simplifie la construction des diagrammes historiques. Les diagrammes sont par la suite produits par la template *createHistoricForGroups*. Ces deux templates sont présentées dans les encadrés ci-dessous.

```

<xsl:template name="createTempHistoricForGroups">
    <xsl:param name="fileName"/>
    <xsl:param name="aliasesNodes"/>
    <xsl:param name="isHousing" select="0"/>
    <xsl:param name="simulationYears" select="20"/>
    <xsl:result-document href="{ $fileName }">
        <root>
            <title>Evolution historique de la taille des
            <xsl:choose>
                <xsl:when test="$isHousing = 0">
                    <xsl:text>groupes sociaux</xsl:text>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:text>types d'habitats</xsl:text>
                </xsl:otherwise>
            </xsl:choose>
            </title>
            <xsl:for-each select="$aliasesNodes">
                <xsl:variable name="color">
                    <xsl:call-template name="getColorOnScale">
                        <xsl:with-param name="colorNumber"
select="position()"/>

```

```

                                <xsl:with-param name="numberColors"
select="count($aliasesNodes)"/>
                                </xsl:call-template>
                                </xsl:variable>
                                <xsl:variable name="initLocalX">
                                <xsl:choose>
                                    <xsl:when test="position() < $maxGroupsForLine">
                                        <xsl:value-of select="$initXHistoric +
($maxLengthHistoric*1.01)*((position() - 1))"/>
                                    </xsl:when>
                                    <xsl:otherwise>
                                        <xsl:value-of select="$initXHistoric +
($maxLengthHistoric*1.01)*((position() mod $maxGroupsForLine))"/>
                                    </xsl:otherwise>
                                </xsl:choose>
                                </xsl:variable>
                                <xsl:variable name="initIteration">
                                <xsl:for-each select="./Iteration">
                                    <xsl:if test="position() = 1">
                                        <xsl:value-of select="@name"/>
                                    </xsl:if>
                                </xsl:for-each>
                                </xsl:variable>
                                <xsl:variable name="initLocalY1">
                                <xsl:choose>
                                    <xsl:when test="starts-with(string($initIteration), 'Init')">
                                        <xsl:value-of select="$initYHistoric"/>
                                    </xsl:when>
                                    <xsl:otherwise>
                                        <xsl:choose>
                                            <xsl:when test="starts-with(string($initIteration),
'Validation')">
                                                <xsl:value-of select="$initYHistoric +
($widthHistoric*($simulationYears+1))"/>
                                            </xsl:when>
                                            <xsl:otherwise>
                                                <xsl:value-of select="$initYHistoric +
($widthHistoric*(number($initIteration) + 1))"/>
                                            </xsl:otherwise>
                                        </xsl:choose>
                                    </xsl:otherwise>
                                </xsl:choose>

```

```

        </xsl:choose>
        </xsl:otherwise>
    </xsl:choose>
</xsl:variable>
<xsl:variable name="initLocalY">
    <xsl:choose>
        <xsl:when test="position() &lt; $maxGroupsForLine ">
            <xsl:value-of select="$initLocalY1"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:variable name="times" select="(position() - (position()
mod $maxGroupsForLine)) div $maxGroupsForLine"/>
            <xsl:value-of select="$initLocalY1 +
$times*($widthHistoric*(number($simulationYears)+5))"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:variable>
<NameNode name="{@id}" xPosition="{ $initLocalX}"
yPosition="{ $initLocalY}" />
    <xsl:variable name="fileNameEnd">
        <xsl:choose>
            <xsl:when test="$isHousing = 0">
                <xsl:value-of
select="'households/socialGroups.xml'"/>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="'hosings/habitats.xml'"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>
    <xsl:for-each select="./Iteration">
        <xsl:variable name="inputFileName">
            <xsl:choose>
                <xsl:when test="starts-with(string(@name),
'Init')">
                    <xsl:value-of
select="concat($rootDirectory, '/init/', $fileNameEnd)"/>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:choose>

```

```

                                <xsl:when test="starts-
with(string(@name), 'Validation')">
                                <xsl:value-of
select="concat($rootDirectory, '/Validation/' , $fileNameEnd)"/>
                                </xsl:when>
                                <xsl:otherwise>
                                <xsl:value-of
select="concat($rootDirectory, '/Iteration_', @name , '/', $fileNameEnd)"/>
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:variable>
                                <xsl:variable name="groupName" select="./@groupName"/>
                                <xsl:variable name="groupNode"
select="document($inputFileName)//Group[starts-with(string($groupName), @id) and ends-
with(string($groupName), @id)]"/>
                                <xsl:if test="$groupNode">
                                <HNode size="{ $groupNode/@totalNumber }"
xPosition="{ $initLocalX }" yPosition="{ $initLocalY + position()*$widthHistoric }" color="{ $color }"/>
                                </xsl:if>
                                </xsl:for-each>
                                </xsl:for-each>
                                </root>
                                </xsl:result-document>
</xsl:template>

```

Encadré 36. Template xsl de création de diagrammes historiques des groupes sociaux et des types d'habitats. Ce template fait partie de la feuille de style xsl Statistics.xsl

```

<xsl:template name="createHistoricForGroups">
    <xsl:param name="tempFile"/>
    <xsl:param name="outputFile"/>
    <xsl:param name="simulationYears"/>
    <xsl:param name="numberAliases"/>
    <xsl:variable name="rootNode" select="document($tempFile)/root"/>
    <xsl:variable name="nameNodes" select="$rootNode//NameNode"/>
    <xsl:variable name="hNodes" select="$rootNode//HNode"/>
    <xsl:variable name="maxLength">
    <xsl:for-each select="$hNodes">
        <xsl:sort select="@size" order="descending" data-type="number"/>
        <xsl:if test="position() = 1">
            <xsl:value-of select="@size"/>

```

```

</xsl:if>
</xsl:for-each>
</xsl:variable>
<xsl:result-document href="{ $outputFile }">
  <svg zoomAndPan="enable">
    <svg id="titleBar" x="0" y="0" width="1500" height="45" viewBox="0 0 1500 45">
      <rect x="0" y="0" width="100%" height="100%"
style="fill:lightgrey;stroke:none" />
      <text x="20" y="18" style="font-family:Times,Helvetica,sans-serif;font-
size:18px;font-weight:bold;pointer-events:none">
        <xsl:value-of select="$rootNode/title"/>
      </text>
    </svg>
    <xsl:call-template name="writeScale">
      <xsl:with-param name="numberAliases" select="$numberAliases"/>
      <xsl:with-param name="simulationYears" select="$simulationYears"/>
    </xsl:call-template>
    <xsl:for-each select="$hNodes">
      <xsl:variable name="width" select="0.5 * $maxLengthHistoric * @size div
$maxLength"/>
      <xsl:variable name="height" select="0.8 * $widthHistoric"/>
      <g transform="translate({@xPosition},{@yPosition})" display="inline">
        <rect x="0" y="0" width="{ $width }" height="{ $height }"
style="fill: {@color}"/>
        <text x="{ $width }" y="{ $height }" style="font-
family:Times,Helvetica,sans-serif;font-size:9px;font-weight:bold;pointer-events:none"><xsl:value-of
select="@size"/></text>
      </g>
    </xsl:for-each>
    <xsl:for-each select="$nameNodes">
      <g transform="translate({@xPosition},{@yPosition})" display="inline">
        <text x="0" y="0" style="font-family:Times,Helvetica,sans-serif;font-
size:15px;font-weight:bold;pointer-events:none"><xsl:value-of select="@name"/></text>
      </g>
    </xsl:for-each>
  </svg>
</xsl:result-document>
</xsl:template>

```

Encadré 37. Template xsl de création de diagrammes historiques des groupes sociaux et des types d'habitats. Ce template fait partie de la feuille de style xsl Statistics.xsl

## A6.3 Tableaux de migration

Les transformations nécessaires pour produire des tableaux récapitulatifs de migration (cf. section 10.2.1.3.) sont programmés sous forme de *templates* de XSL dans la feuille de style *Migrations.xsl*. Les tableaux de migrations sont produits en format html par les templates *buildMigrationTableForGroup* et *buildGeneralMigrationTableForIteration*. La première de ces templates construit le tableau récapitulatif des migrations pour un groupe social donné dans une année de simulation donnée. La deuxième template construit un tableau récapitulatif des migrations pour une année de simulation donnée. Ces deux templates sont présentées ci-après.

```
<xsl:template name="buildMigrationTableForGroup">
  <xsl:param name="groupId"/> <!-- groupe pour lequel on construit le tableau de migrations -->
  <xsl:param name="iteration"/> <!-- année de simulation pour laquelle on construit le tableau de
migrations -->
  <xsl:variable name="migrationNodes" select="document(concat($rootDirectory, '/Market_',
string($iteration), '/Migrations.xml'))"/>
  <xsl:variable name="urbanSectors" select="$groupsAliases//UrbanSector"/>
  <a name="{ $groupId }"><div align="center"> <b>Matrice de migrations pour le groupe
  <xsl:call-template name="getLabelForGroup">
    <xsl:with-param name="groupName" select="$groupId"/>
    <xsl:with-param name="aliases" select="$groupsAliases"/>
  </xsl:call-template>
pour l'année : <xsl:value-of select="1973+$iteration"/>-<xsl:value-of select="1974+$iteration"/>
  </b>
  </div></a>
  <br/>
  <a href="#indexGroups">Retourner à l'index</a>
  <br/>
  <table>
    <tr bgcolor="#5ebbc7">
      <th rowspan="1" align="center">Origines</th>
      <th rowspan="1" align="center"
colspan="{count($urbanSectors)+1}">Destinations</th>
    </tr>
    <tr bgcolor="#5ebbc7">
      <th rowspan="1" align="center"></th>
      <xsl:for-each select="$urbanSectors">
```

```

        <th rowspan="1" align="center"><xsl:value-of select="@name"/></th>
    </xsl:for-each>
    <th rowspan="1" align="center">Totaux</th>
</tr>
<xsl:for-each select="$UrbanSectors">
    <xsl:variable name="from" select="@name"/>
    <xsl:variable name="fromNodes" select="$migrationNodes//Migration[ends-
with(@socialGroup, $groupId) and ends-with(@originArea, $from)]"/>
    <tr bgcolor="#ffffaa">
        <td bgcolor="#40838c" align="left">
            <font face="verdana, geneva, arial" size="2" color="#ffffff">
                <xsl:value-of select="@name"/>
            </font>
        </td>
        <xsl:for-each select="$UrbanSectors">
            <xsl:variable name="to" select="@name"/>
            <th rowspan="1" align="center"><xsl:value-of
select="sum($fromNodes[ends-with(@destinationArea, $to)]/@size)"/></th>
        </xsl:for-each>
        <th rowspan="1" align="center"><xsl:value-of
select="sum($fromNodes/@size)"/></th>
    </tr>
</xsl:for-each>
<tr bgcolor="#ffffaa">
    <xsl:variable name="fromNodes" select="$migrationNodes//Migration[ends-
with(@socialGroup, $groupId) and ends-with(@originArea, 'OUTSIDE')]/>
    <td bgcolor="#40838c" align="left">
        <font face="verdana, geneva, arial" size="2" color="#ffffff">
            Nouveaux arrivants
        </font>
    </td>
    <xsl:for-each select="$UrbanSectors">
        <xsl:variable name="to" select="@name"/>
        <th rowspan="1" align="center"><xsl:value-of
select="sum($fromNodes[ends-with(@destinationArea, $to)]/@size)"/></th>
    </xsl:for-each>
    <th rowspan="1" align="center"><xsl:value-of
select="sum($fromNodes/@size)"/></th>
</tr>
<tr bgcolor="#5ebbc7">

```



```

        <td align="left">
            <font face="verdana, geneva, arial" size="2" color="#ffffff">
                Totaux
            </font>
        </td>
        <xsl:for-each select="$UrbanSectors">
            <xsl:variable name="to" select="@name"/>
            <th rowspan="1" align="center"><xsl:value-of
select="sum($migrationNodes//Migration[ends-with(@socialGroup, $groupId) and ends-
with(@destinationArea, $to)]/@size)"/></th>
            </xsl:for-each>
            <th rowspan="1" align="center"><xsl:value-of
select="sum($migrationNodes//Migration[ends-with(@socialGroup, $groupId)]/@size)"/></th>
        </tr>
    </table>
</xsl:template>

```

Encadré 38. Template xsl de création de tableaux récapitulatifs de migrations. Ce template fait partie de la feuille de style xsl Migrations.xsl

```

<xsl:template name="buildGeneralMigrationTableForIteration">
    <xsl:param name="iteration"/><!-- année de simulation pour laquelle on construit le tableau de
migrations -->
    <xsl:variable name="migrationNodes" select="document(concat($rootDirectory, '/Market_',
string($iteration), '/Migrations.xml'))"/>
    <xsl:variable name="urbanSectors" select="$groupsAliases//UrbanSector"/>
    <a name="{ $iteration }"><div align="center"> <b>Matrice générale de migrations pour l'année :
<xsl:value-of select="1973+$iteration"/>-<xsl:value-of select="1974+$iteration"/>
</b>
</div></a>
    <br/>
    <a href="#indexGroups">Retourner à l'index</a>
    <br/>
    <table>
        <tr bgcolor="#5ebbc7">
            <th rowspan="1" align="center">Origines</th>
            <th rowspan="1" align="center"
colspan="{count($urbanSectors)+1}">Destinations</th>
        </tr>
        <tr bgcolor="#5ebbc7">
            <th rowspan="1" align="center"></th>
            <xsl:for-each select="$urbanSectors">
                <th rowspan="1" align="center"><xsl:value-of select="@name"/></th>
            </xsl:for-each>
        </tr>
    </table>

```

```

        </xsl:for-each>
        <th rowspan="1" align="center">Totaux</th>
    </tr>
    <xsl:for-each select="$SurbanSectors">
        <xsl:variable name="from" select="@name"/>
        <xsl:variable name="fromNodes" select="$migrationNodes//Migration[ends-
with(@originArea, $from)]"/>
        <tr bgcolor="#ffffaa">
            <td bgcolor="#40838c" align="left">
                <font face="verdana, geneva, arial" size="2" color="#ffffff">
                    <xsl:value-of select="@name"/>
                </font>
            </td>
            <xsl:for-each select="$SurbanSectors">
                <xsl:variable name="to" select="@name"/>
                <th rowspan="1" align="center"><xsl:value-of
select="sum($fromNodes[ends-with(@destinationArea, $to)]/@size)"/></th>
            </xsl:for-each>
            <th rowspan="1" align="center"><xsl:value-of
select="sum($fromNodes/@size)"/></th>
        </tr>
    </xsl:for-each>
    <tr bgcolor="#ffffaa">
        <xsl:variable name="fromNodes" select="$migrationNodes//Migration[ends-
with(@originArea, 'OUTSIDE')]"/>
        <td bgcolor="#40838c" align="left">
            <font face="verdana, geneva, arial" size="2" color="#ffffff">
                Nouveaux arrivants
            </font>
        </td>
        <xsl:for-each select="$SurbanSectors">
            <xsl:variable name="to" select="@name"/>
            <th rowspan="1" align="center"><xsl:value-of
select="sum($fromNodes[ends-with(@destinationArea, $to)]/@size)"/></th>
        </xsl:for-each>
        <th rowspan="1" align="center"><xsl:value-of
select="sum($fromNodes/@size)"/></th>
    </tr>
    <tr bgcolor="#5ebbc7">
        <td align="left">

```

```

        <font face="verdana, geneva, arial" size="2" color="#ffffff">
        Totaux
        </font>
    </td>
    <xsl:for-each select="$urbanSectors">
        <xsl:variable name="to" select="@name"/>
        <th    rowspan="1"    align="center"><xsl:value-of
select="sum($migrationNodes//Migration[ends-with(@destinationArea, $to)]/@size)"/></th>
        </xsl:for-each>
        <th    rowspan="1"    align="center"><xsl:value-of
select="sum($migrationNodes//Migration/@size)"/></th>
    </tr>
</table>
</xsl:template>

```

Encadré 39. Template xsl de création de diagrammes historiques des groupes sociaux et des types d'habitats. Ce template fait partie de la feuille de style xsl Statistics.xsl

# BIBLIOGRAPHIE

- Abraham A. et Ramos V. (2003), « Web usage mining using artificial ant colony clustering and linear genetic programming ». In *The Congress on Evolutionary Computation*, pages 1384–1391, Canberra, Australia, 08-12 Decembre : IEEE-Press.
- Abu Lughod J, (1969), « Testing the theory of Social Area Analysis : The Ecology of Cairo (Egypt) », In *American Sociological Review*, 34, pp. 189-212.
- Alegre G., Jullien G. et Oberkampf E. (1975), *Le modèle de Forrester*. Dossier d'études, DEA, Université Paris X – Nanterre, 49 p.
- Alexandrov D. (2000), « Randomized algorithms for the minmax diameter k-clustering problem ». In *Proceedings of ECCO 13*, pp. 193–194, Capri, Italy, Mai.
- Almeida C.M., Monteiro A.M.V., Câmara G., Soares-Filho B.S., Cerqueira G.C., et Pennachin C.L. (2002), « Modeling urban land use dynamics through Bayesian probabilistic methods in a cellular automaton environment ». In *Proceedings of the 29th International Symposium on Remote Sensing of the Environment*, Buenos Aires, Argentina, 8–12 Avril.
- Antcliff, S. (1993). *An Introduction to DYNAMOD – A Dynamic Population Microsimulation Model*. Canberra, Australia: National Centre for Social and Economic Modelling.
- Auriac F., (1979) *Système économique et espace*, Thèse d'Etat sur la schématisation systémique de l'évolution du vignoble languedocien.
- Azzag H, Picarougne F., Guinot C., Venturini G. (2004), « Un survol des algorithmes biomimétiques pour la classification ». In M. Chavent et M. Langlais (dirs.) *Classification et fouille de données. RNTI-C-1*, ed. Cepaduès, pp. 13–24.
- Badariotti D., Weber C. (2002), « La mobilité résidentielle en ville. Modélisation par automates cellulaires et système multi-agents à Bogotá ». In *L'espace géographique* 2002/2 : pp. 97 – 108.
- Banos A., Marilleau N., Thevenin T., Chardonnel S., Lang C. et Mas-Boffet A.(2006a). « Concevoir des agents pour simuler les dynamiques urbaines : de l'enquête ménage à la génération de programmes d'activités. », In *Int. Conf. on Spatial Analysis and GEomatic, SAGEO'06*, Strasbourg, France, septembre.

- Banos A., Marilleau N., Thevenin T., Chardonnel S., Lang C. et Mas-Boffet A. (2006b). *Miro, Rapport d'étape n°2 Année 2005-2006 Génération d'une population « synthétique » d'agents*. Rapport technique, PREDIT.
- Bao H.T, Diday E. et Gettler-Summa M. (1988), « Generating Rules for Expert System from Observation ». In *Pattern Recognition Letters*, pp. 265-271.
- Bao H.T, Huyen T., (1991), « A Method for Generating Rules from Examples and its Application ». In *Symbolic and Numeric Data Analysis and Learning*, ed. E. Diday, Nova Science, pp. 493-504.
- Barros J. et Alves JR. (2003), « Simulating Rapid Urbanisation in Latin American Cities ». In *Advanced Spatial Analysis: The CASA Book of GIS*, ed. P. Longley and M. Batty. Londres : ESRI Press.
- Barros J. (2004), *Urban Growth in Latin American Cities. Exploring urban dynamics through agent-based simulation*. Thèse de doctorat. University College London University of London
- Batty M. et Xie Y. (1994), « From cells to cities ». In *Environment and Planning B*, 21: pp 31–48.
- Batty M. (1998), « Urban evolution on the desktop: simulation with the use of extended cellular automata ». In *Environment and Planning A* 30: pp. 1943-1967.
- Batty M., Xie Y., Sun Z. (1999), « Modeling urban dynamics through GIS-based cellular automata ». In *Computers, Environment and Urban Systems* 23: pp. 205-253.
- Batty M. (2001), « Polynucleated Urban Landscapes », In *Urban Studies*, Vol.38, No.4, pp. 635-655.
- Benenson I. (1998), « Multi-agent simulations of residential dynamics in the city ». In *Computers, Environment and Urban Systems* 22 (1): pp. 25-42.
- Benenson I., Omer I., Hatna E. (2002), « Entity-based modeling of urban residential dynamics: The case of Yaffo, Tel Aviv ». In *Environment and Planning B: Planning and Design* 29(4): pp. 491 – 512
- Benenson, I. & Torrens, P.M. (2003) « Geographic Automata Systems: A New Paradigm for Integrating GIS and Geographic Simulation ». In *Proceedings of the 6th AGILE Lyon, France*, 24 – 26 Avril.

- Benenson, I. & Torrens, P.M. (2004a) *Geosimulation: Automata-Based Models of Urban Phenomena*. Chichester: John Wiley & Sons.
- Benenson, I. & Torrens, P.M. (Eds.) (2004b), « Geosimulation: Object-Based Modelling of Urban Phenomena ». Special Issue, *Computers, Environment and Urban Systems*, 28 (1/2).
- Berry B.J.L., Kasarda J.D. (1977), *Contemporary urban-ecology*. Macmillan Publ. Co : New-York, 497 p.
- Bertalanffy L. (1968), *General System Theory, Foundations, Development, Applications*, New York: George Braziller.
- Bertelle C., Jay V., Lerebourg S., Olivier D. et Tranouez P. (2002), « Dynamic clustering for auto-organized structures in complex fluid flows ». In *ESS 2002 Conference*, Dresden (Allemagne), Octobre.
- Besussi E., Cecchini I. et Rinaldi E. (1998) « The diffusion of the Italian North-east: Identification of Urban Dynamics using Cellular Automata urban models » in *Computer, Environment and Urban Systems* Vol. 22. N.5, pp. 497 – 523.
- Boman M., Holm E. (2005), « Multi-agent Systems, Time Geography, and Microsimulations ». In M.-O. Olsson and G. Sjöstedt (eds) *Systems Approaches and Their Application*. Springer Netherlands, pp 95-118.
- Bonnefoy J. L. (2002). « From households to urban structures: space representations as engine of dynamics in multi-agent simulations ». In *Cybergéo, Revue Européenne de Géographie*
- Bonnet M. & Desjeux D., (2000), *Les territoires de la mobilité*, PUF.
- Bonté L. (2005), *Représentation multi-échelle pour plates-formes à grands nombres d'agents*. Mémoire de master de Recherche – Intelligence artificielle. USTL – LIFL Université de Lille, Juin.
- Boudon R. (1977), *Effets pervers et ordre social*. Paris : PUF.
- Bourdieu P. (1980), *Le sens pratique*. Minuit.
- Bourdieu P. (1972), *Esquisse d'une théorie de la pratique*. Seuil.
- Brener N., Ben Amar F., Bidaud P. (2004), « Analysis Of Self Reconfigurable Modular Systems A Design Proposal For Multi Modes Locomotion ». In *IEEE Int. Conf. Robotics & Automation (ICRA'04)*, New Orleans, LA.

- Brown D.G., Page S.E., Riolo R. et Rand, W. (2004), « Agent Based and Analytical Modeling to Evaluate the Effectiveness of Greenbelts ». In *Environmental Modelling and Software*.
- Brown D.G., Page S., Riolo R., Zellner M. et Rand W. (2005), « Path dependence and the validation of agent-based spatial models of land use ». In *International Journal of Geographical Information Science* Vol. 19, No. 2, Février, pp. 153–174
- Brunet R. (1990), « La dynamique des systèmes » chapitre IX du livre premier *Le déchiffrement du monde* de BRUNET R. & DOLLFUSS O., Mondes nouveaux, Geo Universelle, vol. 1, Belin-reclus.
- Bura S., Guérin-Pace F., Mathian H., Pumain D., Sanders L. (1996). « Multi-agents systems and the dynamics of a settlement system ». In *Geographical Analysis*, vol. 28, n° 2
- Burgess & Park (1926), *The urban community*, University of Chicago Press.
- Casti, J. L. (1997), « Can you trust it? On the reliability of computer simulation models ». In *Complexity* 2 (5): pp. 8-11.
- Caldwell S. et Keister L.A. (1996), « Wealth in America: family stock ownership and accumulation 1960–95 ». In: Clarke (1996).
- Clarke G.P., ed. (1996), « Microsimulation for Urban and Regional Policy Analysis ». In *European Research in Regional Science* vol 6, Londres, pp. 88–116.
- Clarke M. et Wilson A.G. (1986). « A framework for dynamic comprehensive urban models: the integration of accounting and Microsimulation approaches ». In *Sistemi Urbani* 213: pp. 145–177.
- Clarke K.C., Hoppen S., Gaydos L. (1997). « A self-modifying cellular automaton model of historical urbanization in the San Francisco Bay area ». In *Environment and Planning B: Planning and Design*, volume 24, pp 247 – 261.
- Colonna A., DiStephano V., Lombardo S., Papini L. et Rabino A. (1998). « Learning Cellular Automata : modelling urban modelling ». In *Proceedings of the 3rd International conference on GeoComputation*.
- Colomi A., Dorigo M. et Maniezzo V. (1991). « Distributed optimization by ant colonies ». In *Proceedings of the First European Conference on Artificial Life*, pp. 134–142.
- Coquillard P. et Hill D. (1997), *Modélisation et simulation d'écosystèmes : des modèles déterministes aux simulations à évènements discrets*, Masson.

- Corkill D. D. (1991), « Blackboard Systems ». In *AI Expert*, 6(9):40-47, Septembre.
- David K. (1996), « A Methodology and Modelling Technique for Systems of BDI Agents », In *the seventh European Workshop on Modelling Autonomous Agents in a Multiagents world*, Eindhoven, Hollande.
- Daudé E. (2002), *Modélisation de la diffusion d'innovations par la simulation multiagents. L'exemple d'une introduction en milieu rural*. Thèse de doctorat, Université d'Avignon, 327P.
- Deneubourg J.-L., Goss S., Franks N.R., Sendova-Franks A., Detrain C. et Chretien L. (1990), « The dynamics of collective sorting: robot-like ant and ant-like robots ». In *Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pp. 356–365.
- Derycke P.-H. et Franck B. (1977), « Simulation et jeux urbains – une application de la théorie des automates cellulaires à la simulation du développement spatial de la ville ». In *Mélanges offerts à Henri GUITTON*.
- Diday E. (1991), *Des Objets de l'Analyse de Données A Ceux de l'Analyse des Connaissances. Inductions symboliques et numériques à partir des données-* ed par Kodratoff Y et Diday E., Cépaduès.
- Diday E. (1992), « Eléments d'Analyse des Données Symboliques ». In *PRC IA Apprentissage Symbolique et Numérique*. Marseille, décembre.
- Dauphiné A. (2003), *Les théories de la complexité chez les géographes*, Economica, 248 p.
- Dorigo M. (1992), *Optimization, Learning and Natural Algorithms*. Thèse de doctorat, Politecnico di Milano, Italie.
- Drogoul A. (1993), *De la simulation multi-agents à la résolution collective de problèmes*, Thèse de doctorat, Université Paris VI.
- Drogoul A. et Ferber J. (1992). « Multi-Agent Simulation as a Tool for Modeling Societies: Application to Social Differentiation in Ant Colonies ». In *Proceedings of MAAMAW'92*, Viterbo.
- Drogoul A. et Ferber J. (1994). « Multi-agent simulation as a tool for studying emergent processes in societies ». In N. Gilbert et J. Doran (éds.), *Simulating societies : the computer simulation of social phenomena*, pp 127-142, Londres : UCL Press



- Dubois M., Le Guyadec Y., Duhaut D.(2003), « Control of Interconnected Homogeneous Atoms: Language and Simulator » In *Proceedings of Clawar 2003* Catana (Italie)
- Duhaut D. (2002), « Robotic Atom ». In *Proceedings of Clawar 2002*, 24-26 september Paris (France)
- Dureau, F., Dupont V., Lelièvre E., Lévy J.-P. & Lulle T., Eds. (2000). *Métropoles en mouvement: une comparaison internationale*. Collection Villes, Anthropos-IRD. 656 p.
- Efron B. & Tibshirani R. (1994). *An Introduction to the Bootstrap*. Chapman & Hall/CRC.
- Epstein J.M. et Axtell R.L. (1996). *Growing Artificial Societies. Social Science From the Bottom Up*. Brookings Institution Press MIT Press.
- Epstein J.M. (1999) « Agent-Based Computational Models and Generative Social Science ». In *Complexity*, Vol.4-5: pp. 41-60.
- Ferber J. (1995), *Les systèmes multi-agents – Vers une intelligence collective*. Paris : InterEditions, 522p.
- Ferber J. (1999), *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison Wesley Longman.
- Forrester J. W. (1968), *Principles of Systems*, 1st preliminary edn. MIT Press, Cambridge, MA.
- Forrester J.W. (1969), *Urban Dynamics*. MIT Press, Cambridge, MA
- Forrester J.W. (1971), « Counterintuitive behavior of social systems ». In *Technology Review*. Alumni Association of the Massachusetts Institute of Technology
- Forrester J.W. (1979), *Dynamiques Urbaines*. Traduit de l'anglais par Patrick Sylvestre-Baron, , Paris : Economica.
- Fritzke, B.(1994), « Fast learning with incremental RBF networks ». In *Neural Processing Letters*, 1(1):-5.
- Gilbert N. et Troitzsch K. G. (1999), *Simulation for the Social Scientist*. Buckingham-Philadelphia : Open University Press.
- Gil-Quijano J., Piron M. (2007a.), « Formation automatique de groupes d'agents sociaux par techniques d'apprentissage non supervisé ». In *Actes de EGC'07 Atelier Fouille de Données et Algorithmes Biomimétiques*. Namur Belgique, 23 Janvier.

- Gil Quijano J., Piron M., Drogoul A. (2007b.) « Mechanisms of automated formation and evolution of social-groups: A multi-agent system to model the intra-urban mobilities of Bogotá city ». Chapitre 12 du livre *Social Simulation : Technologies, Advances and New Discoveries*. B. Edmonds, C. Hernandez, K. Troitzsch (eds.) European Social Simulation Association ; Idea Group Inc. USA (à paraître).
- Gil Quijano J., Piron M., Drogoul A., (2007c.), « Vers une simulation multi-agent de groupes d'individus pour modéliser les mobilités résidentielles intra-urbaines » in *Dynamiques urbaines et mobilités, Numéro spécial de la Revue Internationale de Géomatique*, volume 17, n°2 pp. 161-181.
- Gimblett R. (2002), *Integrating GIS and agent-based modelling techniques for simulating social and ecological processes*, Londres : Oxford University Press, 327 p.
- Grafmeyer Y., Joseph I., (1990) (2ème éd.), *L'école de Chicago: naissance de l'écologie urbaine*. Aubier : Paris, 378 p.
- Grafmeyer Y. (1994) *Sociologie urbaine*, Nathan, Collection Université : Paris.
- Grassé P.-P. (1959), « La reconstruction du nid et les coordinations inter-individuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. La théorie de la Stigmergie : Essai d'interprétation du comportement des termites constructeurs », In *Insectes Sociaux*, numéro 6, pp. 41-80.
- Guermond Y. (1984), *Analyse des systèmes en géographie*, collection Sciences des systèmes, Presses Universitaires de Lyon.
- Hägerstrand T. (1975a). « Space, time and human condition ». In: *Dynamic Allocation of Urban Space*, ed. by Anders Karlqvist, Lars Lundqvist, and Folke Snickars. Farnborough: Saxon House, pp. 2–12.
- Hägerstrand T. (1975b). « Survival and arena: on the life-history of individuals in relation to their geographical environment ». *Monadnock* 49: pp. 9–29.
- Hägerstrand T. (1995). « Action in the physical everyday world ». In: *Diffusing Geography: Essays for Peter Haggett*, ed. by A.D. Cliff, P. Gould, A. Hoare, and N. Thrift. Blackwell.
- Hamming R. (1950), « Error-detecting and error-correcting codes » In *Bell System Technical Journal* 29(2):147-160
- Holm E. et Sanders L. (2007), « Spatial microsimulation models ». In: *Models in Spatial Analysis*, Lena Sanders ed.. Iste Publishing Company, pp 159-196.

- Holm E., Holme K., Lindgren U., Mäkilä K. (2002), « The SVERIGE Spatial Microsimulation Model : Content, validation, and example applications ». In *GERUM kulturgeografi*. Umeå, Sweden: CERUM Kulturgeografi, Kulturgeografiska Institutionen/SMC, Umeå University.
- Holmström J. (2002) *Growing Neural, GasExperiments with GNG, GNG with Utility and Supervised GNG*. Thèse de master en informatique. Uppsala University – Department of Information Technology Computer Systems. Uppsala Suède
- Hornik K., Stinchcombe M. et White H. (1989), « Multilayer feedforward networks are universal approximators », In *Neural Networks*, vol. 2, no. 5, pp. 359-366.
- Houssein B., Chaib-draa B., Kropf P. (2001), « Multiagent auctions for multiple items ». In *Proceedings of the Third International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2001)*, Montreal, 28 Mai.
- Huhns M.H., et Singh M.P. (1998), *Readings in Agents*. Publié par Morgan Kauffman, 523p.
- Isnard H., Racine JB, Reymond H. (1981), *Problématiques de la géographie*, PUF.
- Kennedy J. et Eberhart R. C. (1995), « Particle swarm optimization ». In *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ. pp. 1942-1948.
- Kirtland D., Gayos L., Clarke K., Decola L., Acevedo W. et Bell C. (1994). « An analysis of human induced land transformations in the San Francisco Bay/Sacramento Area ». In *World Ressources Review*, 6(2) : 206-217.
- Kohler T. A. (2000), « Putting social sciences together again: An introduction to the volume ». In T. A. Kohler and G. J. Gumerman, editors, *Dynamics in Human and Primate Societies: Agent-Based Modeling of Social and Spatial Processes*. New York : Oxford University Press.
- Kohler T. A., Kresl J., Van West C., Carr E. et Wilshusen R. H. (2000). « Be there and then: a modeling approach to settlement determinants and spatial efficiency among late ancestral pueblo populations of the Mesa Verde region, US SouthWest ». In T. A. Kohler and G. J. Gumerman, editors, *Dynamics in Human and Primate Societies: Agent-Based Modeling of Social and Spatial Processes*. New York : Oxford University Press.
- Kohonen, T. (1982), « Self-organized formation of topologically correct feature maps », In *Biol. Cybern.*, vol 43, pp. 59-69.

- Krupp H.-J. (1986), « Potential and limitations of Microsimulation models ». In: *Microanalytic Simulation Models to Support Social and Financial Study*, ed. by Guy H. Orcutt, J. Mertz, and H. Quinke. North-Holland.
- Kuntz P., Layzell P. et Snyers D. (1997), « A colony of ant-like agents for partitioning in vlsi technology ». In P. Husbands et I. Harvey, editors, *Proceedings of the Fourth European Conference on Artificial Life*, pp. 417–424.
- Laarhoven P. et Aarts P. (1987), *Simulated Annealing Theory and Applications*. Boston : D. Reidel Publishing Company.
- Labroche N., Monmarché N., et Venturini G. (2002), « A new clustering algorithm based on the chemical recognition system of ants ». In F. van Harmelen, editor, *Proceedings of the 15th European Conference on Artificial Intelligence*, pp. 345–349, Lyon, France: IOS Press. Juillet.
- Lalage A. (2001) *Desarrollo de Agentes Geograficos en SIG*. Mémoire de Magister en Informatique dirigé par German Bravo. Universidad de los Andes. Bogotá Colombie.
- Lam N. S.-N. et Quattrochi D.A. (1992), « On the issues of scale, resolution, and fractal analysis in the mapping sciences ». In *The Professional Geographer*, 44, pp. 88-98.
- Lebart L., Piron M., Morineau A. (2006) *Statistique exploratoire multidimensionnelle : visualisation et inférence en fouilles de données*. 4<sup>ème</sup> édition. Paris : Dunod, 464 p.
- Lelièvre E., Levy-Vroelant C. (ed.), (1992). *La ville en mouvement : habitat et habitants*. Paris : L'Harmattan, Collection Villes et entreprises, 357 p.
- Leontief W. (1951). *The Structure of the American Economy, 1919–1939*. Oxford University Press.
- Lévy, J.-P. & Dureau F., Eds. (2002). *L'accès à la ville. Les mobilités spatiales en questions*. Collection Habitat et Société. Paris : L'Harmattan. 411 p.
- Looney C.G. (1997), *Pattern Recognition Using Neural Networks. Theory and Algorithms for Engineers and Scientists*. Oxford : Oxford University Press.
- Lumer E.D. et Faieta B. (1994), « Diversity and adaptation in populations of clustering ants ». In *Proceedings of the Third International Conference on Simulation of Adaptive Behaviour*, pp. 501–508.

- Machnik L. (2006), « Documents clustering method based on Ants Algorithms ». In *Proceedings of the International Multiconference on Computer Science and Information Technology*, pp. 123 – 130.
- Maes, P., (ed.) (1990). *Designing Autonomous Agents: Theory and Practice From Biology to Engineering and Back*. Cambridge, MA: MIT Press.
- Malenfant J., Denier S. (2003), « ARM Un modèle Réflexif Asynchrone Pour Les Objets Répartis Et Réactifs ». In *Revue des Sciences et Technologies de l'Information Vol(9) 1-2*.
- Marilleau N., Lang C., Chatonnay P., Philippe L. (2005a) « Une bibliothèque multi-agents pour la simulation de la mobilité intra-urbaine », In *procs. of the Int. Conf. on Spatial Analysis and GEomatic, SAGEO 2005*, Avignon, France, Juin.
- Marilleau, N., Lang, C., Chatonnay, P., et Philippe, L.(2005b). « A Meta-Model of Group for Urban Mobility Modeling ». In *procs. of The Third Int. Conf. on Active Media Technology, AMT 2005*, Takamatsu, Japon, pages 397--400, Mai.
- Milligan G.W., Cooper M.C. (1985), « An Examination of Procedures for Determining the Number of Clusters in a Data set ». In *Psychometrika*, vol. 50, pp. 159-179.
- Minsky M.L. et Papert S.A. (1988), *Perceptrons*, Expanded Edition, Cambridge, MA : MIT Press.
- Möhring M. (1996), « Social Science Multilevel Simulation with MIMOSE ». In K.G. Troitzsch, U. Mueller, G.N. Gilbert and J.E. Doran, *Social Science Microsimulation* Berlin: Springer
- Monmarché N., Slimane M., et Venturini G. (1999), « On improving clustering in numerical databases with artificial ants ». In D. Floreano, J.D. Nicoud, et F. Mondala, editors, *5th European Conference on Artificial Life (ECAL'99), Lecture Notes in Artificial Intelligence, volume 1674*, pp. 626–635, Swiss Federal Institute of Technology, Lausanne, Switzerland : Springer-Verlag. 13-17 Septembre
- Monmarché N., Guinot C., et Venturini G. (2002), « Fouille visuelle et classification de données par nuage d'insectes volants ». In *RSTI-RIA-ECA: Méthodes d'optimisation pour l'extraction de connaissances et l'apprentissage*, (6): pp.729–752.
- Morin, E. (1990), *Introduction à la pensée complexe*, Le Seuil.

- Morineau A. (1984), « Note sur la caractérisation statistique d'une classe et les valeurs-tests », In *Bull. Techn. du Centre de Statistique et d'Informatique Appliquées*, 2, pp. 20-27.
- Morineau A., Summa M., Tong H. (1995), « Marquage Sémantique des Classes et des Axes ». In *28èmes journées de l'ASU*, Paris, pp. 468-472.
- Mullon C., Piron M, Treuil J.P. 2001. « An agent-based approach of urban migration flows ». in *ESS'2001*, Marseille, 380-385.
- Nakache J.P., Confais J. (2005), *Approche pragmatique de la classification*. Paris : Editions Technip.
- Ohsumi N. (1988), « Role of computer graphics in interpretation of clustering results ». In *Recent Developments in Clustering and Data Analysis*, Diday E. et al. (eds) Boston : Academic Press.
- Orcutt, G.H. (1957). « A new type of socio-economic system ». In *Review of Economics and Statistics* (58) : pp. 773–794.
- Orcutt G. H., Greenberger M., Korbel J., et Rivlin A. (1961), *Microanalysis of Socioeconomic Systems: A Simulation Study*. Harper & Row.
- Orcutt G.H., Caldwell S. Wertheimer et R. (1976), *Policy Exploration Through Microanalytic Simulation*. Washington DC : Urban Institute.
- O'Sullivan, D. (2001), « Graph-cellular automata: a generalised discrete urban and regional model ». In *Environment and Planning B: Planning & Design* 28(5), pp. 687–705.
- Paelinck J.H.P. (1972), « Modeles urbains dynamiques: Etude critique » In *Revue économique*, Vol. 23, No. 6 (novembre), pp. 931-951.
- Pécaut D. (2003), *Violencia y Política, Ensayos sobre el conflicto colombiano*, Medellin : Editora Hombre Nuevo, Universidad del Valle, 164 p.
- Parias A. (2001), *Segmentación del mercado de vivienda y segregación socio-espacial en Bogotá*, Thèse de doctorat, Institut d'Urbanisme de Paris, Université de Paris XII-Val-de-Marne.
- Park, Burgess & Mackensie (1925), *The city*, University of Chicago Press.
- Park R.E. (1929), « The city as a Social Laboratory ». In Smith & White, *An Experiment in Social Science Research* , University of Chicago Press.

- Parker D., Manson S. M., Janssen M., Hoffmann M. J. et Deadman, P. (2003), « Multi-Agent Systems for the Simulation of Land-Use and Land-Cover Change: A Review ». In *Annals of the Association of American Geographers*, 93 (2), pp. 314-337.
- Pelleg D. et Moore A. (2000), « Xmeans : Extending K-means with efficient estimation of the number of clusters ». In *Procs. Of the 17<sup>th</sup> International Conference on Machine Learning*, pp. 727–734.
- Pinchemel P. & Pinchemel G. (1988), *La face de la Terre. Eléments de géographie*, Armand Colin : Paris.
- Piron M., Dureau F., Mullon C., 2003, « Vers un modèle dynamique des mobilités résidentielles : développement sur Bogota », In *Actes des Sixièmes Rencontres de Théo Quant*, Edités par C. Tannier, H. Houot et S. Chardonnel, Université de Franche-Comté, 9p
- Piron M., Dureau F., Mullon C. (2004), « Utilisation de typologies multi-dates pour l'analyse des transformations socio-spatiales de Bogota », In *Cybergeo: revue européenne de géographie*, n°256, 17p.
- Piron M. (2005), « Comment évaluer et représenter le changement de la structure sociale de Bogotá ? Les niveaux d'observation dans l'analyse du changement », In *Actes des Septième Rencontres de Théo Quant*, Edités par C. Tannier, H. Houot et S. Chardonnel, Université de Franche-Comté, 9p
- Piron M., Dureau F., Mullon C. (2006), « Analyse de la dynamique du parc de logements à Bogotá : utilisation de typologies multidates », In *Cybergeo: revue européenne de géographie*, n°334, 23p.
- Piron M., Mullon C., Dureau F., Deman A. (2007) « Modèle dynamique des mobilités résidentielles intra-urbaines à Bogota : des niveaux de modélisation intermédiaires ». In *L'espace géographique* à paraître le dernier trimestre de 2007.
- Portugali J., Benenson I. (1997), « Individuals' cultural code and residential self-organization in the city space ». In *the second annual conference of GeoComputation '97 & SIRC '97*. University of Otago, New Zeland, 26-29 Août.
- Portugali J., Benenson I. et Omer I. (1997), « Spatial cognitive dissonance and sociospatial emergence in a self-organizing city », In *Environment and Planning B* 24: pp. 263-285.
- Portugali J. (2000), *Self-Organization and the City*. Berlin: Springer-Verlag.

- Préteceille E. et de Queiroz L.D. (1999). « Tendencias da segregação social en metópoles globais e desiguais : Paris e Rio de Janeiro no anos 80 », In *EURE*, n°26, 76.
- Prigogine I. & Stengers I. (1979) *La nouvelle alliance*, Paris : Ed. Folio.
- Proctor G. et Winter C. (1998), « Information flocking: Data visualisation in virtual worlds using emergent behaviours ». In J.-C. Heudin, editor, *Proceedings of the 1st Int. Conf. Virtual Worlds, VW*, volume 1434, pp. 168–176. Springer-Verlag.
- Prud'homme R., De La Brunetiere J., et Dupuy G. (1972), *Les jeux de simulation urbanistiques*, Paris : Tema-Formation, 234 p.
- Pruvot M., Weber-Klein C. (1984), « Ecologie urbaine factorielle comparée : essai méthodologique et application à Stratsbourg ». In *L'Espace Géographique*, n°2, pp. 136-150.
- Pumain D., Sanders L., Saint-Julien T. (1989), *Villes et auto-organisation*, Economica, 189 p.
- Pumain D. (1996) *Penser la ville. Théories et modèles*, Anthropos - thème
- Rand W., Brown D.G., Page S.E., Riolo R., Fernandez L.E. et Zellner M. (2003), « Statistical validation of spatial patterns in agent-based models ». In *Proceedings of Agent Based Simulation*, Montpellier, France.
- Renault V. (2001), *Organisation de sociétés d'agents pour la visualisation d'informations dynamiques*. Thèse de doctorat Université Pierre et Marie Curie- Paris VI
- Reynolds C. W. (1987), « Flocks, herds, and schools: A distributed behavioural model ». In *SIGGRAPH '87 Conference Proceedings*, 21(4): pp. 25–34.
- Roncayolo M. (1990), *La ville et ses territoires*, Gallimard, Collection Folio : Paris.
- Rossenblatt F. (1958), « The perceptron : a probabilistic model for information storage and organization in the brain » In *Psycho. Review*, vol. 654, pp. 386-408.
- Rothenberg J. (1974), « Problems in the modelling of urban development – A review article on -Urban Dynamics by Jay W. Forrester- ». In *Journal of Urban Studies*, vol. 1, No. 1, janvier, pp. 1 – 20.
- Sanders L., Pumain D., Mathian H., Guerin-Pace F., Bura S., 1997, « SIMPOP : a multi-agents system for the study of urbanism ». In *Environment and Planning*, No 24, pp. 287-305.



- Sargent R.G. (1985), « An expository on verification and validation of simulation models », In *Proceedings of the 17th conference on Winter simulation*, pp.15-22, San Francisco, California, United States, Decembre 11-13
- Sargent, R. G. (2001), « Some Approaches and Paradigms for Verifying and Validating Simulation Models », in *Proceedings of the 2001 Winter Simulation Conference*, Disponible en ligne à :  
<http://www.informs-cs.org/wsc01papers/013.PDF>.
- Sargent R.G. (2005), « Verification and validation of simulation models ». In *Proceedings of the 37th conference on Winter simulation*, Orlando, Florida, United States.
- Saura, S. & J. Martínez-Millán. 2001. « Sensitivity of landscape pattern metrics to map spatial extent ». In *Photogrammetric Engineering and Remote Sensing* 67 (9): pp. 1027-1036
- Schelling T.S. (1971), « Dynamic Models of Segregation ». In *Journal of Mathematical Sociology* ;1 pp. 143-186.
- Schlesinger, S., Crosbie R.E., Gagne R.E., Innis G.S., Lalwani C.S., Loch J. (1979), « Terminology for model credibility ». In *Simulation* 34 (3): pp. 101-105.
- Schofisch B., Hadeler K. P. (1996), « Dimer Automata And Cellular Automata ». In *Physica D*, vol. 94, no 4, pp. 188-204
- Semboloni F. 2000. « The growth of an urban cluster into a dynamic self-modifying spatial pattern ». In *Environment and Planning B: Planning & Design*. 27 (4): pp. 549- 564.
- Servat D. (2002), *Modélisation de dynamiques de flux par agents. Application aux processus de ruissellement, infiltration et érosion*. Thèse de doctorat Université Pierre et Marie Curie- Paris VI.
- Shanon C.E. (1949), « A mathematical theory of communication ». In *Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-656, juillet et octobre.
- Shaw, A. (2000), *CORSIM Analyst Documentation*.  
 Disponible à : <http://www.strategicforecasting.com/docs/>
- Shi W., et Pang M.Y.C. (2000), « Development of Voronoi-based cellular automata—an integrated dynamic model for Geographical Information Systems ». In *International Journal of Geographical Information Science*. 14 (5):455-474.

- Schumpeter, J.A. (1954), *History of Economic Analysis*, New York : Oxford University Pres, vol. 3.
- Stern M. et Bursaux F. (1971), « Analyse dynamique de l'évolution urbaine chez J.W. Forrester ». In *Cahiers de l'IAURP*, vol. 23, 24.
- Stone R. (1966), *Mathematics in the Social Sciences and Other Essays*. London: Chapman & Hall.
- Torrens, P.M. (2001), « New tools for simulating housing choices ». In *Program on Housing and Urban Policy Conference Paper Series (C01-006)*. University of California, Berkeley, Institute of Business and Economic Research and Fisher Center for Real Estate and Urban Economics.
- Torrens, P.M. & Benenson, I.(2005) « Geographic Automata Systems ». In *International Journal of Geographic Information Science*, 19(4): pp. 385-412.
- Tinbergen J. (1939) *Statistical Testing of Business Cycle Theories Vol.2, Business Cycles in the United States of America 1919–1932*, Geneva: League of Nations, Economic Intelligence Service.
- Tranouez P., Lerebourg S., Bertelle C. et Olivier D. (2003), « Contribution à la représentation multi-échelle des écosystèmes aquatiques ». In *JFSMA'2003 Conf., Hammamet (Tunisie)*, Novembre 2003. In book : J.-P. Briot and K. Ghédina ed. *Déploiement des systèmes multi-agents: vers un passage à l'échelle*, pp 255-259, TSTI/hors série, Editions Hermès.
- Turner M.G., Gardner R.H. et O'Neill, R.V. (2001), *Landscape Ecology in Theory and Practice: Pattern and Process*, New York: Springer-Verlag.
- Vanbergue D. (2000), « Modélisation de phénomènes urbains : Simulation des migrations intrar-urbaines ». In Presty, S. et Sayettat-Fau, C.(eds). *Systèmes multi-agents*. Hermes.
- Vanbergue D. (2003), Conception de simulation multi-agents. Application à la simulation des migrations intra-urbaines de la ville de Bogotá. Thèse de doctorat. Université Pierre et Marie Curie, Paris VI, Décembre.
- Vicari J. (1981), *Agir sur la ville: Essai d'écologie urbaine.* -\_ ed. du Moniteur : Paris, 276p.
- Wahle J., Neubert L., Esser J., Schreckenberg M. (2001), « A Cellular Automaton Traffic Flow Model for Online Simulation of Traffic ». In *Parallel Computing*, vol. 27 No.(5), pp. 719-735.

- Weber M. (1913), *Über einige Kategorie der verstehenden Soziologie (1913)*. Traduction française dans *Essais sur la théorie de la science*, Paris : Plon.
- Weber M. (1965), *Essais sur la théorie de la science*, traduction par J. Freund, Paris : Plon.
- White R. et Engelen G. (1993). « Cellular Automata and Fractal Urban Form : a Cellular Modelling Approach to the Evolution of Urban Land use Patterns ». In *Environment and Planning A*, vol. 25, pp. 1175-1199.
- White R. et Engelen G. (1997). « Cellular automata as the basis of integrated dynamic regional modelling ». In *Environment and Planning B : Planning and Design*, vol. 24, pp. 235-246.
- Wiener N. (1950), *Cybernetics and Society, the Human Use of Human Beings*, New York : Da Capo, Press.
- Wirth L. (1938), « Urbanism as a way of life », In *American Journal of Sociologie*, vol. 44, juillet.
- Wooldridge M.J. (2002), *An Introduction to MultiAgent Systems*, Wiley, 366p.
- Wu, F., 1998. « An experiment on the generic polycentricity of urban growth in a cellular automatic city ». In *Environment and Planning B-Planning & Design* 25(5), pp. 731-752.
- Youness G. (2004), *Contributions à une méthodologie de comparaison de partitions*. Thèse de doctorat , Université Pierre et Marie Curie – Paris VI.
- Xie Y. et Batty M. (2005), « Integrated Urban Evolutionary Modeling ». In *GeoDynamics* (edited by Peter Atkinson, Giles Foody, Steven Darby and Fulong Wu), Boca Raton: CRC Press, pp. 273-294.

<b>INTRODUCTION .....</b>	<b>1</b>
<b><u>1. INTRODUCTION GÉNÉRALE.....</u></b>	<b>2</b>
1.1. <i>La ville un système complexe façonné par les mobilités résidentielles.....</i>	<i>4</i>
1.2. <i>La recherche en systèmes multi-agents .....</i>	<i>5</i>
1.3. <i>Application sur Bogotá.....</i>	<i>6</i>
<b><u>2. ENJEUX DANS LA MODÉLISATION MULTI-AGENT.....</u></b>	<b>8</b>
2.1. <i>Génération et manipulation automatique d'agents composés abstraits : Un problème multi-échelle .....</i>	<i>9</i>
2.1.1. <i>La constitution automatique d'agents composés.....</i>	<i>10</i>
2.1.2. <i>Les interactions à différents niveaux et l'évolution des agents composés.....</i>	<i>12</i>
2.2. <i>Un grand nombre d'agents : simulations à large échelle.....</i>	<i>13</i>
<b>PARTIE I: ETAT DE L'ART .....</b>	<b>16</b>
<b><u>CHAPITRE I : MODÉLISATION URBAINE ET SIMULATION .....</u></b>	<b>17</b>
1.1. <i>Introduction aux notions de modélisation et de simulation .....</i>	<i>17</i>
1.2. <i>Modélisation et simulation en géographie .....</i>	<i>20</i>
<b><u>CHAPITRE II : MODÉLISATION INFORMATIQUE DES DYNAMIQUES URBAINES .....</u></b>	<b>21</b>
2.1. <i>Modélisation sur un niveau.....</i>	<i>21</i>
2.2. <i>Modélisation avec plusieurs niveaux.....</i>	<i>24</i>
2.2.1. <i>Approches descendantes : La microsimulation .....</i>	<i>24</i>
2.2.2. <i>Approches ascendantes : Les systèmes auto-organisés.....</i>	<i>27</i>
2.2.2.1. <i>Modèles évolués de microsimulation.....</i>	<i>27</i>
2.2.2.2. <i>La microsimulation spatiale.....</i>	<i>28</i>
2.2.2.3. <i>Les automates cellulaires.....</i>	<i>29</i>
2.2.2.4. <i>Les systèmes multi-agents .....</i>	<i>30</i>
2.3. <i>Conclusion et proposition d'une approche multi-niveaux intermédiaire.....</i>	<i>34</i>
<b><u>CHAPITRE III : LES MÉCANISMES DE FORMATION DE GROUPES .....</u></b>	<b>37</b>
3.1. <i>La classification automatique.....</i>	<i>38</i>
3.1.1. <i>Méthodes classiques de partitionnement (clustering) .....</i>	<i>39</i>
3.1.1.1. <i>Algorithmes agglomératifs de partitionnement .....</i>	<i>40</i>
3.1.1.1.1. <i>Les nuées dynamiques (ou centres mobiles ou k-means).....</i>	<i>40</i>
3.1.1.1.2. <i>Réseaux de neurones artificiels (RNA) concurrents.....</i>	<i>42</i>
3.1.1.2. <i>Algorithmes hiérarchiques de partitionnement.....</i>	<i>47</i>
3.1.1.2.1. <i>Principe de l'algorithme.....</i>	<i>47</i>
3.1.1.2.2. <i>Stratégies de regroupement.....</i>	<i>48</i>
3.1.1.2.3. <i>Choix de la meilleure partition.....</i>	<i>49</i>
3.1.2. <i>Méthodes de partitionnement par systèmes multi-agents .....</i>	<i>49</i>
3.1.2.1. <i>Partitionnement par colonies de fourmis.....</i>	<i>50</i>
3.1.2.2. <i>Algorithmes basés sur le tri de fourmis.....</i>	<i>50</i>
3.1.2.3. <i>Algorithmes basés sur le déplacement de particules (la compétition spatiale) .....</i>	<i>51</i>
3.1.3. <i>Méthodes de validation et interprétation automatique d'une partition.....</i>	<i>53</i>
3.1.3.1. <i>Validation des partitions .....</i>	<i>53</i>
3.1.3.1.1. <i>L'Indice F-Maximum.....</i>	<i>54</i>
3.1.3.1.2. <i>L'Indice de Davis-Bouldin.....</i>	<i>54</i>
3.1.3.2. <i>Interprétation automatique d'une partition.....</i>	<i>54</i>
3.2. <i>Modèles multi-agents de formation d'agents composés .....</i>	<i>56</i>
3.3. <i>Conclusion sur la formation et l'interaction des groupes .....</i>	<i>58</i>
<b><u>CHAPITRE IV : L'ÉVALUATION DES SIMULATIONS MULTI-AGENTS .....</u></b>	<b>61</b>
<b>PARTIE II : MODÈLES DE SIMULATION MULTI-AGENT .....</b>	<b>64</b>
<b><u>CHAPITRE V : INTRODUCTION AU MODÈLE.....</u></b>	<b>65</b>
5.1. <i>Présentation générale.....</i>	<i>65</i>
5.2. <i>Définitions.....</i>	<i>69</i>
<b><u>CHAPITRE VI : LES INTERACTIONS ENTRE GROUPES SOCIAUX DANS LE MARCHÉ DU LOGEMENT, UN MÉCANISME DE MOBILITÉ.....</u></b>	<b>73</b>
6.1. <i>Calcul de la préférence localisée.....</i>	<i>74</i>
6.2. <i>Les agents du système.....</i>	<i>75</i>
6.2.1. <i>L'agent enchère.....</i>	<i>75</i>
6.2.1.1. <i>Exécution d'une enchère.....</i>	<i>77</i>

6.2.1.2. Distribution des logements .....	77
6.2.2. Les agents acheteurs .....	79
6.2.2.1. Groupe social localisé .....	79
6.2.2.1.1. Recherche d'enchères par un groupe localisé .....	80
6.2.2.1.2. Détermination du prix à proposer dans une enchère .....	81
6.2.2.1.3. Réception des logements .....	82
6.2.2.2. Groupe social non localisé .....	84
6.2.2.2.1. Recherche d'enchères .....	85
6.2.2.2.2. Détermination du prix à proposer dans une enchère .....	86
6.2.2.2.3. Réception des logements .....	88
6.3. Le contrôle de l'interaction .....	89
6.3.1. Initialisation du marché .....	90
6.3.1.1. Création des groupes sociaux localisés .....	90
6.3.1.2. Création des groupes sociaux non localisés .....	91
6.3.1.3. Création des enchères .....	92
6.3.2. Gestion de l'interaction .....	93
<b>CHAPITRE VII : MODÈLES DE GESTION DE LA DISTRIBUTION DE LA POPULATION .....</b>	<b>96</b>
7.1. Modèle synthétique de gestion de la distribution de la population .....	97
7.1.1. Représentation de la distribution de la population .....	97
7.1.2. Prise en compte de la mobilité .....	98
7.1.3. Evolution de la population de ménages et du parc de logements .....	99
7.1.3.1. Evolution des groupes sociaux .....	100
7.1.3.2. Evolution des types d'habitat .....	101
7.2. Modèle étendu de gestion de la distribution de la population .....	103
7.2.1. Représentation de la distribution de la population .....	105
7.2.2. Prise en compte de la mobilité .....	107
7.2.2.1. Décision de déménager au niveau mésoscopique .....	108
7.2.2.2. Décision de déménager au niveau mésoscopique avec une participation « faible » du niveau microscopique .....	109
7.2.3. Evolution au niveau microscopique .....	109
7.2.3.1. Règles d'évolution pour les ménages .....	110
7.2.3.2. Règle d'évolution pour les logements .....	110
7.2.3.3. Exécutions d'une règle d'évolution .....	111
7.2.3.4. Nombre d'entités traitées lors d'UNE exécution d'une règle d'évolution .....	111
7.2.3.5. Construction et exécution des règles d'évolution .....	112
7.2.3.5.1. Règles de suppression (disparition de la distribution initiale) .....	112
7.2.3.5.2. Règles de création (création de la distribution finale) .....	113
7.2.4. Reconstitution des groupes sociaux et des types d'habitats .....	114
7.2.5. Histoire des groupes sociaux et des types d'habitat .....	117
<b>CHAPITRE VIII : ALGORITHMES DE FORMATION AUTOMATIQUE DE GROUPES .....</b>	<b>120</b>
8.1. Classification automatique basique .....	121
8.2. Adaptation des cartes auto-organisatives .....	122
8.3. Multi-Kmeans .....	125
8.4. Algorithme basé sur les nuages de particules .....	126
8.4.1. Propriétés des particules .....	128
8.4.1.1. Différence de comportements entre agents .....	128
8.4.1.2. Espace d'interaction .....	128
8.4.1.3. Force d'attraction/répulsion .....	128
8.4.1.4. Energie du système .....	130
8.4.1.5. Localisation initiale .....	130
8.4.2. Algorithme de classification des agents .....	134
8.4.2.1. Constitution des groupes .....	135
8.4.2.2. Détection des groupes .....	136
8.4.2.3. Optimisation de l'algorithme .....	138
8.4.2.4. Prise en compte de l'évolution de l'ensemble de micro-entités .....	139
<b>CHAPITRE IX : MÉCANISME DE VALIDATION .....</b>	<b>141</b>
9.1. Calcul de la corrélation .....	142
9.2. Validation visuelle .....	145
9.3. Validation interne .....	146
<b>PARTIE III APPLICATION ET RÉSULTATS .....</b>	<b>147</b>
<b>CHAPITRE X : PRODUCTION DES RÉSULTATS .....</b>	<b>149</b>
10.1. Fichiers de résultats .....	149
10.1.1 Description du fichier socialGroups.xml .....	149
10.1.2 Description du fichier habitats.xml .....	152

10.1.3 Description du fichier Migrations.xml.....	154
10.1.4 Fichier GroupsAliases.xml.....	155
<b>10.2. Traitement des résultats</b> .....	<b>157</b>
10.2.1. Production de la cartographie des résultats .....	158
10.2.1.1. Cartes de distribution spatiale .....	159
10.2.1.2. Cartes de validation .....	160
10.2.1.3. Production de tableaux récapitulatifs de migration.....	160
10.2.2. Description automatique des groupes sociaux et des types d'habitat.....	161
10.2.3. Production des diagrammes historiques de groupes et de types d'habitat .....	163
<b>10.3. Résultats de validation</b> .....	<b>164</b>
<b>CHAPITRE XI : APPLICATION À BOGOTA</b> .....	<b>165</b>
<b>11.1. Objectifs des Expériences</b> .....	<b>165</b>
<b>11.2. Description des Expériences</b> .....	<b>166</b>
11.2.1 Description des données .....	166
11.2.1.1. Populations considérées.....	166
11.2.1.2. Cartes de secteurs urbains .....	168
11.2.2. Prétraitement des données .....	170
11.2.3. Paramétrage des simulations .....	172
<u>11.2.3.1. Paramètres de mobilité (voir 11.1.)</u> .....	172
11.2.3.1.1. Stratégies de mobilité.....	172
11.2.3.1.2. Evolution de l'urgence pour résoudre la demande.....	172
<u>11.2.3.2. Paramètres de formation des groupes</u> .....	173
11.2.3.2.1. Niveau d'abstraction élevé.....	173
11.2.3.2.2. Niveau d'abstraction faible .....	174
<u>11.2.3.3. Paramètres de construction des règles d'évolution</u> .....	174
11.2.3.3.1 Niveau d'abstraction faible .....	175
11.2.3.3.2 Niveau d'abstraction élevé.....	175
11.2.3.4. Paramétrage de référence.....	176
11.2.4. Description des types de simulations effectuées .....	177
11.2.4.1. Simulations avec le modèle synthétique.....	177
11.2.4.2. Simulations avec le modèle étendu.....	178
11.2.4.2.1. Localisation du processus de décision.....	178
11.2.4.2.2. Paramétrage initial.....	178
11.2.4.3. Récapitulatif des différents types de simulations effectués.....	178
<b>11.3 Résultats des expériences</b> .....	<b>179</b>
11.3.1. Résultats préliminaires .....	179
11.3.1.1. Stabilité du mécanisme de mobilité et précision des résultats.....	179
11.3.1.2. Sensibilité des résultats par rapport à l'algorithme de classification utilisé dans la formation de groupes .....	181
11.3.1.3. Choix du modèle et de l'algorithme de classification.....	184
11.3.1.4. Conclusion des résultats préliminaires .....	186
11.3.2. Détermination du paramétrage.....	187
11.3.2.1. Paramètres de mobilité .....	187
11.3.2.1.1. Analyse de la sensibilité des résultats par rapport à la stratégie de déménagement.....	187
11.3.2.1.2. Analyse de sensibilité par rapport à l'évolution de l'urgence pour résoudre la demande de groupes-sociaux-non-localisés .....	189
11.3.2.2. Paramètres de formation de groupes.....	190
11.3.2.3. Paramètres de construction des règles d'évolution .....	191
11.3.3. Résultats graphiques .....	192
<b>11.4 Discussion sur les résultats des simulations</b> .....	<b>193</b>
<b>PARTIE IV DISCUSSION</b> .....	<b>196</b>
<b>ANNEXES</b> .....	<b>206</b>
<b><u>ANNEXE I. DESCRIPTION DES CARACTÉRISTIQUES DES MÉNAGES ET LOGEMENTS DE LA VILLE DE BOGOTA</u></b> .....	<b>207</b>
<b><u>ANNEXE II. REPRÉSENTATION DES PROFILS</u></b> .....	<b>209</b>
<b><u>ANNEXE III. CALCULS DE DISTANCES ENTRE PROFILS</u></b> .....	<b>210</b>
A3.1. Distance de Hamming.....	210
A3.2. Distance Euclidienne .....	213
<b><u>ANNEXE IV. MESURES DE CORRÉLATION DE DEUX ENSEMBLES DE DONNÉES</u></b> .....	<b>215</b>
A4.1. Corrélacion croisée normalisée.....	216
A4.2. Corrélacion croisée normalisée centrée.....	216
A4.3. Corrélacion de Moravec.....	216
<b><u>ANNEXE V. EXEMPLES DE RÉSULTATS GRAPHIQUES DES SIMULATIONS EFFECTUÉES</u></b> .....	<b>218</b>

<i>A5.1. Diagrammes historiques</i> .....	218
<i>A5.2. Diagrammes de caractérisation automatique des groupes</i> .....	220
A5.2.1. Diagrammes de caractérisation automatique des groupes sociaux .....	220
A5.2.2. Caractérisation automatique des types d'habitats.....	223
<i>A5.3. Cartes de validation de la distribution des groupes sociaux et de l'occupation des types d'habitat</i> .....	228
A5.3.1. Cartes de validation de la distribution spatiale des groupes sociaux pour l'année finale de simulation .....	229
A5.3.1. Cartes de validation de l'occupation des types d'habitat pour l'année finale de simulation.....	233
<b><u>ANNEXE VI. FEUILLES XSL POUR LA TRANSFORMATION DES RÉSULTATS D'UNE SIMULATION</u></b> .....	239
<i>A6.1. Production de cartes</i> .....	239
A6.1.1. Cartes de distribution.....	240
A6.1.2. Cartes de validation .....	246
<i>A6.2 Diagrammes et tableaux statistiques</i> .....	256
A6.2.1. Production de diagrammes de description automatique des groupes sociaux et des types d'habitat.....	256
A6.2.2. Production de tableaux récapitulatifs de description automatique des groupes sociaux et des types d'habitat .....	261
A6.2.3. Production de diagrammes historiques des groupes sociaux et des types d'habitats.....	264
<i>A6.3 Tableaux de migration</i> .....	269
<b>BIBLIOGRAPHIE</b> .....	274