

# T.P. Simulation (S.M.A.)

Q. MARTY et P. POMERET-COQUOT

Automne 2019

## 1 Introduction

Les commerces en ville se situent généralement autour des grands axes, dans une situation de forte concurrence.

La simulation que nous proposons se base sur l'hypothèse que cette répartition s'explique par le seul comportement de la clientèle. Nous arrivons à reproduire ce mécanisme.

La section 2 présente notre cas d'étude. La section 3 résume les choix de modélisation détaillés dans la fiche *ODD*. La section 4 expose nos résultats et constatations. La conclusion est en section 5.

## 2 Cas d'étude

Nous nous intéressons à la localisation des commerces concurrentiels en ville.

Nous avons observé in-situ que les commerces sont principalement situés sur les axes de passage, et que les commerces proches proposent souvent des services identiques.

Nous souhaitons simuler ces comportements sur des données réelles, et dans le cas de données modifiées (ajout d'un arrêt de bus par exemple).

Nous travaillons d'abord sur la *route de Narbonne* (Toulouse) et les rues environnantes (figure 1), puis vérifions si nous obtenons des comportement similaires sur d'autres quartiers (données sur github).

## 3 Choix de modélisation

La fiche *ODD* accompagnant ce rapport donne les détails de la modélisation. Nous reprenons ici les grandes lignes.

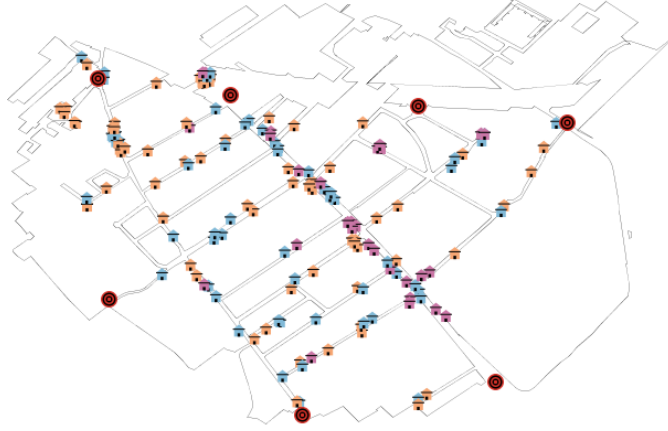


Figure 1: État initial du système (route de Narbonne)

### 3.1 Le monde

Les rues (*roads*) sont les seuls patches accessibles du monde. Nous utilisons les données du cadastre pour les identifier.

Certains patches de rue sont des puits (*wells*), qui servent d'entrée et de sortie pour les clients. Ces puits sont placés manuellement sur la carte (avec QGis) aux extrémités des rues, et possèdent un poids représentant leur fréquentation. On peut en ajouter dynamiquement en cliquant sur la carte.

### 3.2 Les boutiques

#### 3.2.1 Représentation

Les boutiques possèdent une position, un domaine d'activité (*market*) représenté par leur couleur, des fonds (*funds*) représentés par leur taille, et une file d'attente (*queue*).

#### 3.2.2 Initialisation

Les boutiques sont générées d'après la base Sirene : leur position et leur domaine d'activité est réaliste. Toutes commencent avec la même somme d'argent et une file d'attente vide (figure 1)

### 3.2.3 Comportement


Les boutiques gagnent 1 sou lorsqu'un client y consomme, et en perdent lorsqu'un client est généré. Cela nous permet de simuler une taxe qui est exactement égale à l'argent gagné par les commerces, afin de conserver un niveau constant d'argent en circulation.

A chaque client, la file d'attente grandit, elle réduit à chaque tick.

De plus, si une boutique prospère (deux fois ses fonds de départ), alors elle finance un nouveau commerce voisin de même type.

## 3.3 Les clients

### 3.3.1 Représentation

Les clients possèdent de l'argent (*money*), un besoin (*need*) correspondant au *market* d'une boutique et une destination (l'un des puits). 

### 3.3.2 Initialisation

Les clients commencent sur un puits, avec un besoin et une destination (choisis au hasard pondéré). Initialement, ils possèdent tous la même quantité d'argent (*base-money*).

### 3.3.3 Comportement

Un client consomme s'il passe à proximité d'une boutique adaptée, où l'attente est raisonnable. Il donne alors 1 sou à la boutique.

Un client atteignant sa destination disparaît, et un autre client est alors généré. Cela assure une population constante.

## 4 Résultats de la simulation

### 4.1 Paramètres représentatifs

Après stabilisation du système (figure 2), nous observons la répartition géographique des commerces via : la distance au puits le plus proche, la densité locale de client et le facteur d'agglomération concurrentielle (nombre de boutiques de même type à faible distance).

Dans tous les cas suivants, les commerces s'agglomèrent par type (comportement émergent issu des faillites et créations successives).

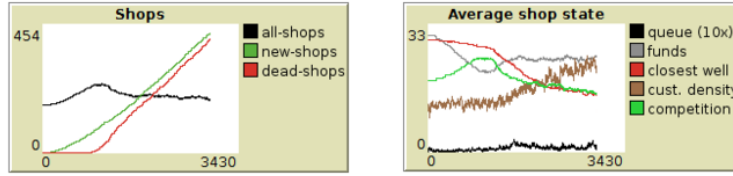


Figure 2: Le nombre de commerces (à gauche), la densité de clients et le taux de concurrence (à droite, courbes marron et vertes) se stabilisent.

Avec une seule consommation par client ( $base-money = 1$ ), et une vitesse d'écoulement de la file d'attente raisonnable ( $queue-speed = 0.1$ ), les commerces se regroupent autour des puits (figure 3).

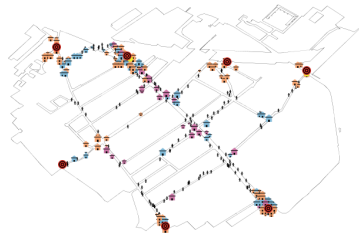


Figure 3: Les clients ne consomment une seule fois

Avec plus d'argent par client ( $base-money = 3$ ) ou des files d'attente moins rapides ( $queue-speed = 0.01$ ), les clients sont poussés à consommer plus loin : les commerces se répartissent sur les axes principaux (figure 4).

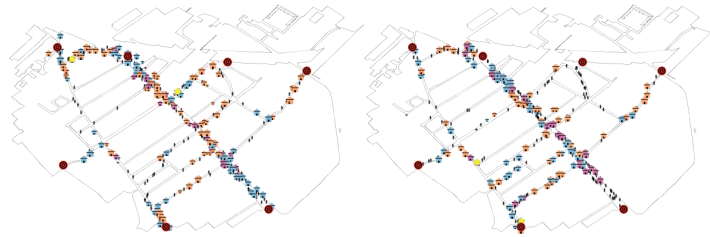


Figure 4: Les clients consomment 3 fois (à gauche), l'écoulement des files d'attentes est lent (à droite)

## 4.2 Ajout de nouvelles sources de clients

Nous pouvons ajouter de nouvelles sources (puits) de clientèle dynamiquement, par un simple clic sur la carte. Après un temps de stabilisation, nous retrouvons le même comportement que précédemment.

Par exemple, reprenons le cas  $base-money = 1$ ,  $queue-speed = 0.1$  (les commerces s'agglomèrent autour des sources).

1. Nous laissons le système se stabiliser comme précédemment
2. Nous ajoutons une source dans la rue *Lefranc de Pompignan* (tout à gauche de la carte).
3. Nous laissons à nouveau le système se stabiliser (figure 5)

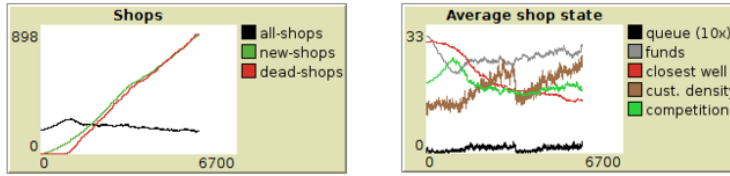


Figure 5: Comportement du système après l'ajout d'une nouvelle source à  $t=3000$  : le système est déstabilisé, mais se redresse

Nous constatons que certains commerces se sont agglomérés à proximité de la nouvelle source (figure 6). En effet les clients issus de cette nouvelle source ont consommé dans les commerces les plus proches, qui ont provoqué de nouvelles créations de commerces. Après plusieurs cycles de création, un groupe s'est finalement installé à proximité de la nouvelle source.

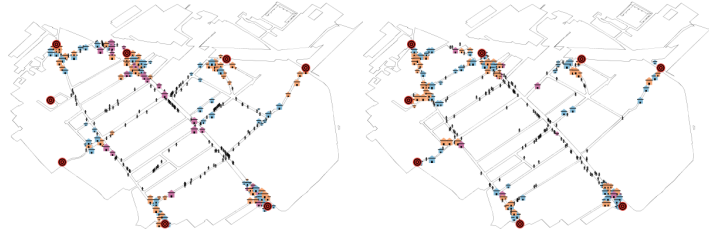


Figure 6: Une source est ajoutée (à gauche), puis les commerces s'en rapprochent (à droite)

L'ajout de nouvelles sources/puits produit ainsi une migration d'une partie des commerces vers cette source. D'une manière similaire, avec les paramètres permettant de peupler les grands axes, l'ajout d'une nouvelle source définit de nouveaux grands axes qui se peuplent eux-aussi (figure 7 : les axes principaux ont changé).

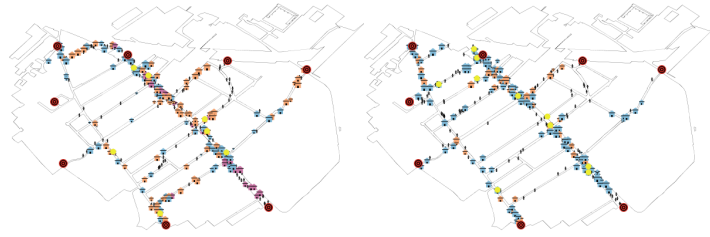


Figure 7: Une source est ajoutée (à gauche), ce qui modifie les axes principaux. Les commerces s'adaptent (à droite)

## 5 Conclusion

Le modèle proposé permet de simuler les comportements concurrentiels des commerces en ville. Nous avons validé notre modèle empiriquement, et regrettons de n'avoir pas pu faire tourner BehaviorSearch sur aucune de nos machines personnelles pour confirmer nos résultats, et ajuster les paramètres avec plus de précision.

Néanmoins, nous parvenons à simuler l'évolution de ces commerces, et cela en permettant d'ajouter dynamiquement de nouvelles sources de clients. Ainsi, nous pouvons estimer l'impact de la création d'une station de bus, métro, etc. sur notre modèle.

## 6 Annexe

Le code (et particulièrement les jeux de données) sont disponibles sur le dépôt *git* hébergé par GitHub, à l'adresse <https://github.com/pPomCo/SMA-M2DC/>