

Modélisation et Simulation MultiAgents

Rapport de Projet

Maxime Desbois et Simon Lassourreille

Décembre 2018

Cette citation dit que vous avez tort (p 34.) elle-ci aussi (p 34.) et celle là que vous êtes moche (p 34.)

I – Hypothèses d'implémentation

$\mathcal{H}1$ – VALEUR D'UN RATING

Dans le modèle, après toute interaction i entre a et b , l'agent a crée un rating r pour chacun des critères c évaluables dans cette interaction. Ce rating prend la forme d'un tuple $r = (a, b, c, i, v)$ où $v \in [-1, 1]$ est la valeur que a donne à b pour le critère c lors de l'interaction i . Cependant, il n'est nulle part précisé comment l'agent a détermine cette valeur. Nous savons que dans le *testbed* fourni par l'article, a reçoit une utilité $\mathbf{ug} \in [-10, 10]$. Nous ferons donc les hypothèses suivantes :

- On peut exprimer v comme une fonction de l' \mathbf{ug} obtenue par a lors de cette interaction, et cette fonction est propre à a :

$$v = \varphi_a(\mathbf{ug})$$

- Au vu du manque d'informations, on fera l'hypothèse que cette fonction normaliser \mathbf{ug} dans l'intervalle de v , et qu'elle est la même pour chaque agent :

$$\varphi_a(x) = \frac{x}{10}, \forall a$$

Cette hypothèse apporte même une autre opportunité d'amélioration au modèle : tous les agents n'ont pas une appréciation linéaire de l'utilité. De même, cette appréciation n'est pas toujours objective. Dans des situations réalistes, certaines personnes notent trop sévèrement, alors que d'autres peuvent être trop indulgentes. Il serait donc intéressant de doter les agents d'utilités non-linéaire dans un premier temps, et peut être par la suite de différencier certains profils d'agents pour intégrer ces disparités dans les notes des différents critères.

$\mathcal{H}2$ – DÉGRADATION DU SERVICE LIÉE À LA DISTANCE

L'article évoque une dégradation linéaire du service lorsque la distance entre les deux agents en interaction, a et b , dépasse la portée opérationnelle $r_o(b)$ de l'agent fournissant le service.

Cependant, les détails de cette dégradation linéaire ne sont pas donnés par l'article. Cette dégradation est donc une fonction de $x = d(a, b)$ la distance entre a et b qui modifie l'utilité u fournie par l'interaction. On peut tout d'abord supposer :

$$f_{UG}(x) = \begin{cases} UG, & \text{si } x \leq r_o(b) \\ UG + \alpha * (x - r_o), & \text{sinon} \end{cases}$$

Avec $\alpha < 0$ pour obtenir une dégradation.

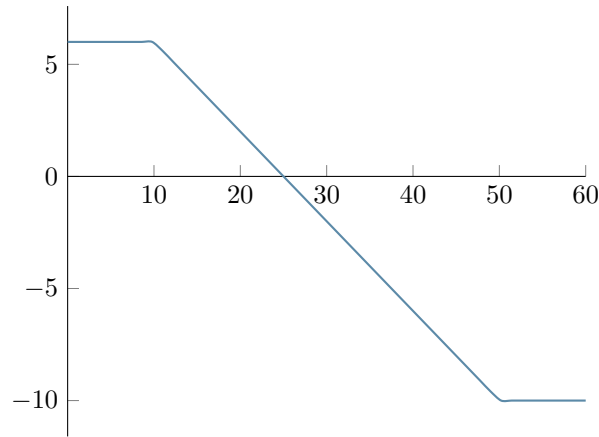
Cependant, il reste quelques problèmes : avec une expression aussi générale, on peut obtenir $u < UG_{min} = -10$, ce qui n'est pas possible. On doit donc introduire une distance maximale à partir de laquelle le service obtenu est minimal, r_{max} . On ajoute donc la condition $f_{UG}(x) = UG_{min}, \forall x \geq r_{max}$. Ayant deux points de cette fonction linéaire, on peut alors déterminer que $\alpha = \frac{UG_{min} - UG}{r_{max} - r_o}$. Ce qui nous donne comme expression finale :

$$f_{UG}(x) = \begin{cases} UG, & \text{si } x \leq r_o(b) \\ UG + \alpha * (x - r_o), & \text{si } r_o \leq x \leq r_{max} \\ UG_{min}, & \text{si } x \geq r_{max} \end{cases}$$

Ou alors, en une seule expression,

$$f_{UG}(x) = \max(UG_{min}, \min(UG, UG + \frac{UG_{min} - UG}{r_{max} - r_o} * (x - r_o)))$$

On peut vérifier avec un graphe que ce comportement correspond à ce qui est attendu, avec : $UG = 6$, $UG_{min} = -10$, $r_o = 10$, $r_{max} = 50$



Cette interprétation nous semble la plus logique. Cependant, il faut choisir une valeur pour r_{max} . Comme ce choix est arbitraire, nous avons choisi $r_{max} = 2 * r_o$.

$\mathcal{H}3$ – RÉPARTITION DES AGENTS DANS L'ESPACE

L'article ne précise pas la loi de probabilité suivie par les coordonnées des agents. C'est un détail important au vu de l'importance de la topologie dans cette simulation. Nous partons du principe que la distribution souhaitée par les auteurs est une distribution uniforme sur l'ensemble de la sphère. Il existe plusieurs méthode pour obtenir une telle distribution.

La première est de considérer les coordonnées sphériques r, φ, θ , avec $r \in [0, 1]$, $\varphi \in [0, \pi[$, $\theta \in]-\pi, +\pi]$. On définit alors la position d'un nouvel agent comme le tirage aléatoire de trois variables uniformes r, φ, θ sur leurs espaces de définitions respectifs.

La seconde consiste à tirer des points uniformes dans un cube en coordonnées cartésiennes : $(x, y, z) \in [-1, 1]^3$, puis rejeter les points qui sont en dehors de la sphère unitaire ($\|(x, y, z)\| > 1$).

Nous avons choisi la première méthode, car elle est plus efficace tout en donnant des résultats corrects.

II – Pseudocode

INITIALISATION : CONSUMER

```

pour chaque consumer a faire
  initialiser une liste vide de taille H
  initialiser r_o
  choisir un niveau d'activité alpha ~ U([0.25, 1.00])
  choisir une position au hasard
fin pour

```

INITIALISATION : PRODUCER

```
pour chaque provider b faire
  déterminer le type de b parmi {good, ok, bad, intermitent}
  initialiser r_o
  si b n'est pas intermitent alors
    initialiser p
    choisir au hasard p dans l'intervale correspondant
  fin si
fin pour
```

DÉROULEMENT D'UN ROUND

```
pour chaque consumer a faire
  déterminer si a nécessite un service
  si vrai alors
    incrémenter i le compteur d'interactions
    b <-- choix d'un provider par a
    ug <-- b génère au hasard une utilité entre -10 et 10
    v <-- déterminer la valeur de l'interaction
    créer un rating r = (a, b, c, i, v)
  fin si
fin pour

# simulation d'instantanéité
pour chaque consumer a faire
  si r existe alors
    enregistrer le rating r dans la liste de a
    envoyer le rating r à b
    b enregistre r si v est meilleur que son moins bon référencement
  fin si
fin pour
```

CHOIX D'UN PROVIDER

```
P <-- récupérer les providers disponibles
pour chaque b dans P faire
  v <-- déterminer la valeur de confiance de b
  si v est défini alors
    insérer b dans la liste NoTrust sinon
    insérer b dans la liste Trust
  fin si
fin pour
trier la liste Trust par valeur de confiance
choisir stratégie (exploration ou exploitation)
si exploration alors
  p <-- choisir un provider au hasard dans NoTrust
sinon
  p <-- provider avec la plus grande valeur de confiance
fin si
```

```
retourner p
```

DÉTERMINER LA VALEUR DE CONFIANCE DE LA COMPOSANTE K

Entrée: RK ensemble de ratings pour le calcul de la composante K

```
 $\Sigma$  <-- 0
 $\Sigma_{vi}$  <-- 0
pour chaque ri dans RK faire
    <-- calculer  $\_K(ri)$ 
     $\Sigma$  <--  $\Sigma$  +
     $\Sigma_{vi}$  <--  $\Sigma$  + * vi
fin pour
 $T_K$  <--  $\Sigma_{vi} / \Sigma$ 
 $\_RK$  <--  $1 - \exp(-K * \Sigma)$ 
 $\Sigma_{vi\_T}$  <-- 0
pour chaque ri dans RK faire
     $\Sigma_{vi\_T}$  <--  $\Sigma_{vi\_T} + \_K(ri) * \text{abs}(vi - T_K)$ 
fin pour
 $\_DK = 1 - 0.5 * (\Sigma_{vi\_T} / \Sigma)$ 
 $\_K = \_RK * \_DK$ 

retourner ( $\_K$ ,  $T_K$ )
```

DÉTERMINER LA VALEUR DE CONFIANCE GÉNÉRALE

```
# Déterminer RI l'ensemble des ratings d'Intéraction
```

III – Etude préliminaire

3.1] Résumez en quelque ligne la problématique des auteurs, et les choix de modélisation qu'ils proposent.

Les auteurs proposent un modèle de confiance permettant à un agent de juger de la qualité des services d'autres agents. Plus précisément, ce modèle doit être décentralisé, robuste aux mensonges.

3.2] Pourquoi ? Test.

IV – Etudes progressive du modèle FIRE

Conception de l'interface NetLogo

Est-ce qu'on en parle ? Si oui, c'est ici !

V – Modèle avec composante IT seule (FIRE-IT)

5.1] Écrivez en pseudo-code l'intégralité des algorithmes que vous allez implémenter pour simuler le modèle FIRE-IT. Expliquez et Justifiez vos choix (notamment pour l'implémentation de la topologie sphérique en Netlogo).

5.2] *Décrivez le protocole expérimental (et votre procédure d'initialisation des agents en particulier), et donnez dans un tableau la valeur de tous les paramètres que vous fixez pour reproduire la Figure 9.*

Initialisation des Agents

Comme discuté dans la section

5.3] *Affichez votre version de la Figure 9 et commentez les résultats.*

VI – Modèle avec composante IT et WR (FIRE-IT-WR)

6.1] *Écrivez en pseudo-code l'intégralité des algorithmes que vous allez implémenter pour simuler le modèle FIRE-IT-WR. Expliquez et Justifiez vos choix. Implémentez en Netlogo.*

6.2] *Indiquez le protocole pour simuler la Figure, et les éventuels changements dans les paramètres (modifications des anciennes valeurs, éventuels ajouts de nouveaux paramètres)*

6.3] *Affichez votre version de la figure 9 complète et expliquez les résultats.*

VII – Modèle avec composante IT, WR et CR (FIRE-IT-WR-CR)

7.1] *Écrivez en pseudo-code l'intégralité des algorithmes que vous allez implémenter l'ajout de la composante CR. Expliquez et Justifiez vos choix. Implémentez en Netlogo.*

7.2] *Indiquez le protocole pour simuler ces Figures, et les éventuels changements dans les paramètres (modifications des anciennes valeurs, éventuels ajouts de nouveaux paramètres).*

7.3] *Affichez vos version des Figures 10 et 11 et expliquez les résultats, notamment en comparaison de la Figure 9.*

VIII – Analyses de sensibilité du modèle FIRE-IT-WR-CR.

8.1] *Affichez un graphique 3D (avec un tableur ou grapheur) qui prend comme axes x et y les poids W_c et W_w et en axe z la performance moyenne des UG. Expliquez comment vous avez obtenu ce graphique et commentez les résultats.*

8.2] *Affichez maintenant (avec les paramètres initiaux de [1]) un histogramme montrant la distribution des valeurs des UGs des clients. Commentez.*

8.3] *Faites varier individuellement les paramètres NPG , NPO et $NB=NPI+NPB$ (mais en respectant toujours $NPG+NPO+NB=100$). Dans des graphes séparés, afficher la moyenne et l'écart type des UG pour chacune 10 ces valeurs. Commentez. Quel est effet de ces paramètres sur la distribution des UG des clients ?*

IX – Extension –

X – Bonus

10.1] *Expliquez en quelques lignes comment le modèle FIRE pourrait être adapté et étendu pour simuler une société telle qu'elle est décrite dans l'épisode Nosedive de la série Black Mirror.*