

## Sujet : modèles “producteur / consommateur” distribués

Il s’agit de modéliser un groupe d’agents pouvant parvenir à un certain équilibre, chaque agent tentant d’optimiser un niveau de satisfaction interne de la manière suivante :

- Tout agent produit une marchandise et en consomme une autre (paramètres fixés à la création de l’agent).
- Tout agent peut acquérir la marchandise qu’il consomme auprès d’un producteur, s’il a de quoi payer <sup>1</sup>.
- Tout agent peut vendre la marchandise qu’il produit s’il trouve un acheteur.
- Le rythme de production est constant jusqu’à atteindre un stock maximal. Lorsque cette valeur maximale est atteinte, l’agent cesse temporairement de produire. Le rythme de consommation est également constant. Les vitesses de production et de consommation sont propres à l’agent, tirées aléatoirement (ou fournies en paramètres) lors de sa création.
- Le niveau de satisfaction de l’agent est à 1 (100%) tant qu’il lui reste un stock de la marchandise qu’il consomme. Il décroît exponentiellement lorsque ce stock est épuisé, jusqu’à ce que l’agent ait pu se réapprovisionner (auquel cas la satisfaction remonte à 1).
- Lorsque l’agent n’est pas satisfait (satisfaction < 50%) et qu’il n’a plus d’argent, il décide de baisser le prix de la marchandise qu’il produit. On prendra 1 pour prix initial de toutes les marchandises.
- Lorsque l’agent est satisfait et qu’il a de l’argent, il décide d’augmenter le prix de cette marchandise produite.

A l’issue d’une simulation, chaque agent affichera son niveau moyen de satisfaction sur la durée de la simulation.

Les agents devant interagir, il est indispensable de se mettre d’accord au préalable sur un protocole de communication. On pourra par exemple utiliser les types de messages suivants :

1. CFP *produit* (Call For Proposal), quand l’agent cherche à acheter un certain produit ;
2. PROPOSE *produit quantité prix*, quand l’agent répond qu’il vend ce produit ;
3. ACCEPT\_PROPOSAL *quantité*, pour demander l’achat d’une certaine quantité de produit ;
4. REJECT\_PROPOSAL, pour indiquer qu’on a trouvé moins cher ailleurs... ;
5. CONFIRM, pour indiquer le transfert effectif de marchandise et d’argent.

On prendra soin de tester son agent en le confrontant à ceux d’autres étudiants, après avoir vérifié qu’il fonctionne correctement avec ses pairs !

On pourra sophisticationner le modèle, par exemple en ajoutant d’autres mécanismes classiques de commerce : messages d’informations sur une baisse de prix, décision d’achat sans besoin réel en prévision d’un risque d’augmentation, etc.

## Détails sur le travail à rendre

Le projet se fait en binômes. À la date indiquée lors du cours vous enverrez une archive nommée d’après les membres du binôme, portant l’une des extensions {zip,tgz} à l’adresse [laroque@u-cergy.fr](mailto:laroque@u-cergy.fr), et vous demanderez confirmation de la réception et de la lisibilité de l’archive.

L’archive contiendra le code source commenté de votre projet, la javadoc produite à partir de vos commentaires et un document d’explications sur la/les façons de lancer votre projet.

---

1. Chaque agent dispose au départ d’une somme égale  $N$ .