

Vie Artificielle

Agents et modèles épidémiques

Dernière mise à jour: janvier 2026

Mots clés: modèle épidémiologique.

Les ressources nécessaires pour cette UE (lien vers les sources, les sujets, etc.) sont disponibles sur Moodle. Pour chaque question, conservez le résultat (code et/ou graphe) afin de pouvoir le montrer.

Modèle SIR

[sur ordinateur]

Téléchargez la source sur Moodle. Vous devez implémenter au cours de ce TME le modèle SIR vu en cours.

QUESTION 1.

Modifiez l'agent *Person* pour que les agents se déplacent au hasard.

Testez votre code avec 500 agents.

QUESTION 2.

Modifiez le code afin que chaque agent est un état de santé. Cet état peut prendre trois valeurs: "S" (sain), "I" (infecté), "R" (guéri). Modifiez le code afin que la couleur de l'agent affichée dépende de son état (S=vert, I=rouge, R=gris). Un agent infecté se déplace deux fois moins vite.

Testez votre code avec une population de 100 agents de chaque type pour tester l'affichage de l'état.

QUESTION 3.

Modifiez le code afin d'implémenter les règles suivantes:

- un agent sain devient infecté s'il se trouve sur la case d'un agent infecté
- un agent infecté guéri après 100 itérations. Un agent infecté se déplace deux fois moins vite.

Testez votre code avec une population de 499 agents sain et 1 agent infecté. Changez la taille de l'environnement pour créer deux situations stables en fin de simulation:

- (1) il n'existe que des agents guéris, et aucun agent sain ou infecté.
- (2) il existe environ le même nombre d'agents sains et d'agents guéris, et aucun agent infecté.

En utilisant le script de génération de graphes vu lors du précédent TME (ou un tableur si vous préférez), vous devrez créer un graphe montrant l'évolution du nombre d'agents sains (ordonnées) au cours du temps (abscisses). Vous devrez montrer un des graphes que vous aurez générés.

QUESTION 4.

On souhaite ensuite implémenter les fonctionnalités suivantes:

- à chaque itération, un agent peut se "reproduire" avec une probabilité $P_{\text{reproduction}}$. C'est à dire qu'un nouvel agent apparaît au même endroit. Cette probabilité par individu est généralement très faible (ex.: 0.001, voire moins).
- chaque agent a une durée de vie maximale (ensuite il meurt et disparaît).
- un agent infecté ou guéri ne peut pas se reproduire.
- un agent sain a une probabilité de tomber malade.

Testez votre code en créant une situation où la population se reproduit suffisamment vite pour contrer l'effet de la maladie. N'hésitez pas à changer la taille de l'environnement.

Remarque: la simulation peut ralentir si vous avez trop d'agents. A vous d'optimiser votre code (en optimisant les suppressions/ajouts d'agents, en réutilisant les objets qui correspondent aux agents supprimés, en mettant une limite artificielle à la taille maximale de la population, etc.). En particulier, voici deux pistes:

- plutôt que de supprimer des agents dans la liste, il est préférable de les désactiver en mettant l'attribut `self.running = False`
- plutôt que de créer de nouveaux agents, il est préférable de réutiliser -- si c'est possible -- les agents inactifs (`self.running = False`)
- plutôt que de parcourir toute la liste des agents pour détecter s'il existe un agent infecté sur la case de l'agent focal, il est préférable de maintenir pour chaque case, les agents présents sur celle-ci [optimisation difficile].

QUESTION 5.

Ecrivez un comportement de fuite pour les agents sains, à exécuter lorsque l'agent détecte un autre agent infecté autour de lui.

Testez votre code en essayant diverses valeurs pour la probabilité qu'à un agent de tomber malade spontanément.

Une fois ce TME terminé, vous pouvez travailler sur le projet. Première étape: intégrer l'écosystème des feux de forêt avec des proies et des prédateurs capable de se reproduire et de mourir, et des virus (nuisibles ou bénéfiques) qui changent les caractéristiques des agents.