



Institut  
Mines-Télécom

# ***La modélisation à base- d'agents avec Netlogo***





## Un dilemme social

- Introduction du dilemme du prisonnier
- Théorie des jeux et économie expérimentale
- structure d'interaction et ségrégation spatiale
- Travail à faire: programme Netlogo du dilemme social



- ❑ Vous travaillez sur un projet joint avec un de vos collègues à rendre pour le lendemain. Il est tard et il est préférable de travailler chacun de votre côté sur chaque partie du projet . Travailler dur ce soir vous coute 10 en termes d'effort et le bénéfice du projet dépendra de l'effort total des deux parties prenantes

| Effort total | 0 | 1  | 2  |
|--------------|---|----|----|
| bénéfice     | 3 | 10 | 18 |

|      |           | Votre collègue |           |
|------|-----------|----------------|-----------|
|      |           | Coopère        | Profiteur |
| Vous | coopère   | 8 8            | 0 10      |
|      | Profiteur | 10 0           | 3 3       |

Le dilemme du Prisonnier



## ❑ **Un dilemme du prisonnier :**

- ✓ Métaphore de situations présentes dans beaucoup de domaine (écologie , science sociale, science politique,...)
- ✓ Jouer des milliers de fois en économie expérimentale dans différents contexte mais
- ✓ Coexistence entre coopérateurs et défecteurs
- ✓ Une observation traditionnelle: coopération conditionnelle
- ✓ Emotion : punir les profiteurs
- ✓ Culpabilité à ne pas coopérer mais idiosyncratique

## Objectif de la modélisation

- ❑ **On introduit la culpabilité  $X$  de profiter de l'autre quand l'autre coopère**
- ❑ **Objectif: reproduire les faits stylisés observés en économie expérimentale**
  - ✓ Coexistence de coopérateur et de profiteur (déflecteur)
  - ✓ Coopération conditionnelle : le nombre de gens prêt à coopérer augmente avec le nombre de coopérateurs
  - ✓ On suppose  $X$  suit une uniforme entre 0 et 1 et est idiosyncratique
  - ✓ On veut aussi montrer par simulation l'émergence de la ségrégation sociale en fonction de la taille du voisinage → des espaces coopératifs et des espaces non coopératifs



**Un dilemme du Prisonnier**



**Le dilemme du Prisonnier modifié (avec culpabilité) X idiosyncratique**



La matrice avec les gains des 2 joueurs  $X_i$  suit une uniforme  $[0 1]$

|          |           | Joueur 2 |           |
|----------|-----------|----------|-----------|
|          |           | Coopère  | défection |
| Joueur 1 | coopère   | 1 - C    | - C       |
|          | défection | 1        | 0         |


|          |           | Joueur 2 |           |
|----------|-----------|----------|-----------|
|          |           | Coopère  | défection |
| Joueur 1 | coopère   | 1 - C    | - C       |
|          | défection | 1 - X    | 0         |

|          |           | Joueur 2               |                           |
|----------|-----------|------------------------|---------------------------|
|          |           | Coopère                | défection                 |
| Joueur 1 | coopère   | 1 - C    1 - C         | - C    1 - X <sub>2</sub> |
|          | défection | 1 - X <sub>1</sub> - C | 0    0                    |



- ❑ une action d'un agent : coopérer ou faire défection
- ❑ Une stratégie d'un agent = une action (coopérer ou faire défection) **étant donné** l'émotion de l'agent
- ❑ Un équilibre : une situation dans laquelle aucun agent n'a une incitation à jouer une autre stratégie : il ne gagnerait pas plus (au plus il pourrait gagner autant)
- ❑ Equilibre:
  - ✓ Faire défection pour chaque joueur quel que soit son émotion est un équilibre

|          |           | Joueur 2               |                           |
|----------|-----------|------------------------|---------------------------|
|          |           | Coopère                | défection                 |
| Joueur 1 | coopère   | 1 - C    1 - C         | - C    1 - X <sub>2</sub> |
|          | défection | 1 - X <sub>1</sub> - C | 0    0                    |



□ **Equilibre:** la stratégie pour chacun des joueurs « Coopérer quel que soit l'émotion » **n'est pas** un équilibre :

✓ Preuve: soit  $\eta$  la probabilité de coopérer ; on cherche à prouver que quel que soit  $X$ ,  $\eta = 1$  est impossible (toujours coopérer indépendamment de son émotion n'est pas un équilibre).

— Le gain espéré de coopérer étant donné  $\eta$  :

- $(1-C) \eta - C(1-\eta)$  (equation 1)

— Le gain espéré de faire défection étant donné  $\eta$

- $(1-X) \eta$

— Un équilibre où chaque joueur coopère quel que soit son  $X$  veut dire

- $\eta = 1$  l'autre joueur coopère avec une probabilité de 1.
- Étant donné que l'autre coopère, je coopère quel que soit mon  $X$

— Ceci implique

- pour tout  $X$ :  $(1-C) \eta - C(1-\eta) > (1-X) \eta$  avec  $\eta = 1$
- Soit  $1-C > 1 - X$ , ce qui n'est pas vrai pour  $X < C$
- Comme  $X$  est distribué uniformément cela arrive avec une probabilité positive pour  $C > 0$ .
- Il existe des  $X$  pour lequel il n'est pas optimal de coopérer et donc la probabilité de tomber sur un tel joueur est positive





## □ Equilibre:

- ✓ Il existe donc un équilibre tel que la stratégie individuelle est
  - faire defection pour  $X$  faible et coopérer pour  $X$  fort .
- ✓ Soit  $\eta$  la probabilité qu'un individu coopère
- ✓ Un individu coopère étant donné  $\eta \neq 0$  si son gain à la coopération étant donné son  $X$  est supérieur à son gain à la defection soit :

$$(1-C) \eta - C (1-\eta) > (1-X) \eta$$

$$\text{soit } X \text{ tel que } X > C / \eta$$

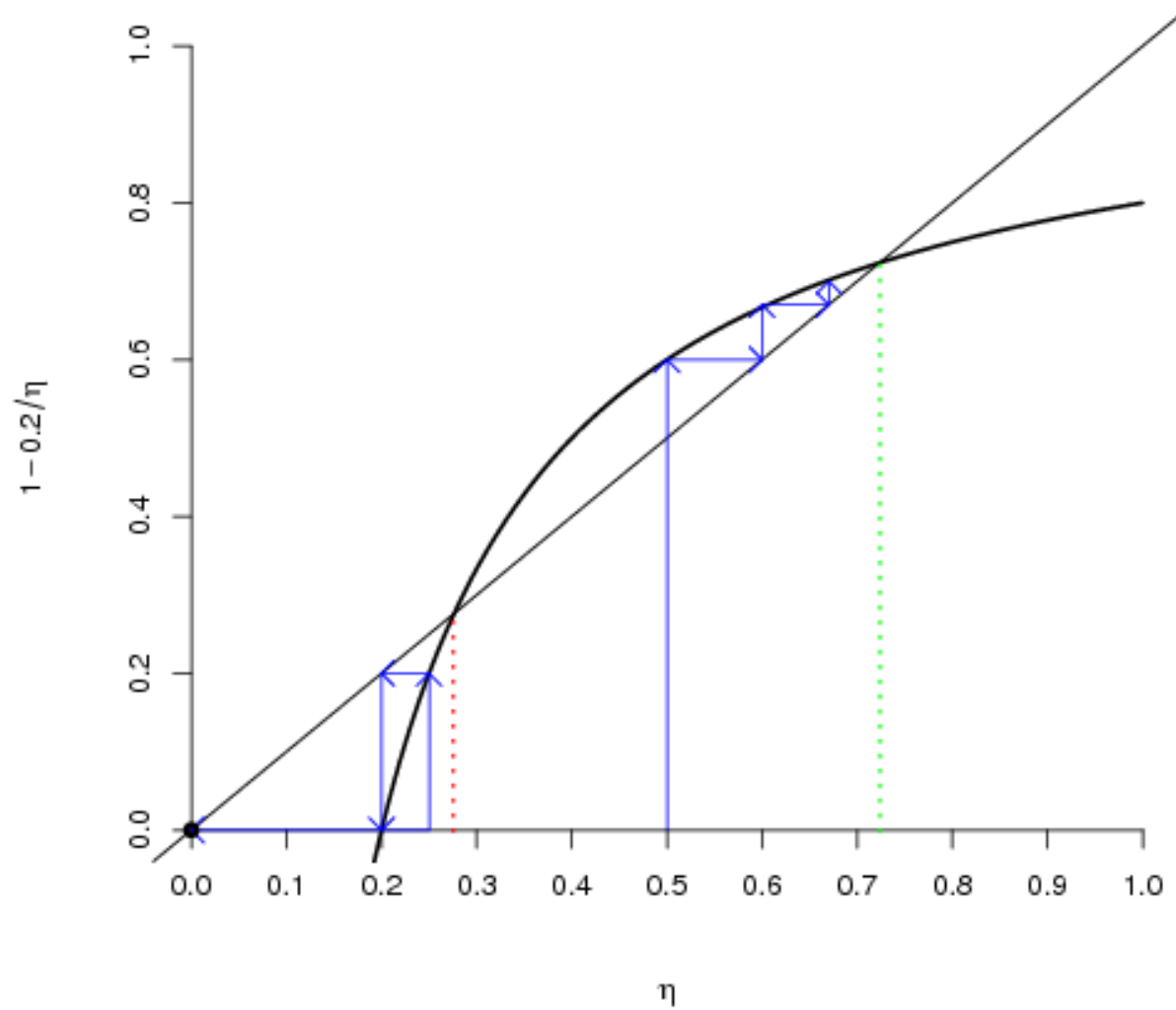
La probabilité que  $X > C / \eta$  est  $1 - F(C / \eta)$

A l'équilibre , la probabilité pour qu'un individu coopère doit correspondre à la probabilité d'avoir une émotion  $X > C / \eta$  soit

$$\eta^* = 1 - F(C / \eta^*)$$

- Si  $X$  est distribué uniformément alors  $\eta^* = 1 - C/n^*$  . Si  $C \leq 0.25$  alors Il y a deux équilibre dont un est instable
- $\eta^*_+ = 0.5 ( 1 + (1- 4C)^{0.5} )$  et  $\eta^*_- = 0.5 ( 1 - (1-4C)^{0.5} )$

## Equilibre pour $C = 0.2$





## □ Le modèle sous netlogo

- ✓ Chaque agent naît avec une émotion distribuée uniformément entre 0 et 1
- ✓ Chaque agent à un voisinage :
  - un paramètre défini dans l'interface permet de choisir la taille ( commune à tous les agents), de ce voisinage : 4 (Von Neumann) , 8 (Moore), 12 , 24 , tous sauf l'agent appelant
- ✓ à chaque tour (pas de temps) un nombre « nb-who-play-this-turn » d'agents joue au dilemme du prisonnier modifié (cf. diapo 7)
- ✓ Chaque agent  $i$  comptabilise la proportion de coopérateurs  $\eta_i$  et défecteurs  $(1 - \eta_i)$  **dans son voisinage** .
- ✓ Il calcule le gain moyen de chaque action étant le taux de coopération dans le voisinage de référence
- ✓ il choisit l'action ayant le plus fort gain moyen



## Pratique de bonne modélisation: le protocole 'ODD' (Overview, Design concepts, and Details)

**Avant de commencer la programmation bille en tête : il s'agit de bien poser le problème et décrire le problème.**

**Bien poser un problème revient à clarifier les aspects suivants (voir le livre Grimm et Railsback 2011):**

### □ Pourquoi de la modélisation

- ✓ Quel est l'objectif général suivi par la modélisation?
  - permet de poser ce qui est important à retenir comme hypothèse
  - Exemple ici :
    - étudier la coopération dans un dilemme social
    - Étudier les effets des structures d'interaction sur la coopération et la ségrégation sociale

## Pratique de bonne modélisation: le protocole 'ODD' (Overview, Design concepts, and Details)

### □ Décrire les Entités , variables, états , échelles

- ✓ Quelles entités sont présentes dans le modèle:
  - groupe / individus: quels variables ou attributs décrivent les individus et les groupes et permettent de les distinguer d'autres groupes/ individus
  - Dans Netlogo:
    - les patches , turtles , breed, patches with , turtle-set,... permet de créer des groupes distincts .
    - exemple de variables: age , sexe, location, fitness, utilité, mémoire
    - netlogo built-in variables , globals, turtles-own,...



## Pratique de bonne modélisation: le protocole The 'ODD' (Overview, Design concepts, and Details)

### □ Décrire les Entités , variables, états , échelles

- ✓ L'espace : type de distance, type de cellule (échelle) et agents sur les cellules (espace résidentiel) , type de monde (torus,...) + descripteur ou variables de l'espace (végétation, altitude,
  - Netlogo: world-width , world-height , patches , in-radius, world wraps horizontally, distance,...
- ✓ L'environnement : caractérise les forces s'appliquant potentiellement à tous les agents (et différentes des conditions spatiales) : Température, regulations fiscales, prix de marché , ...
  - Netlogo: globals , switches, sliders,...

## Pratique de bonne modélisation: le protocole The 'ODD' (Overview, Design concepts, and Details)

### □ Processus , overview et étape:

- ✓ Qui ou quelles entités font quoi et dans quel ordre;
- ✓ quand faut-il actualiser et visualiser les variables d'états.
- ✓ Le temps : variable continu ou variable discrète?
- ✓ Dans quel ordre d'exécution fait on les différents processus ?
- ✓ L'actualisation des différentes variables : synchrone ou asynchrone
- ✓ Présenter le modèle sous forme de pseudo-code pour décrire le déroulement du modèle :
  - **ce code doit servir à d'autres pour ré-implémenter le modèle indépendamment du langage et de la façon de programmer une fonction ou procédure (il peut exister plusieurs façon d'implémenter une fonction ou procédure).**
- ✓ Préciser le contenu de certaines procédures dans des sous-procédures.

## Pratique de bonne modélisation: le protocole The 'ODD' (Overview, Design concepts, and Details)

### □ Conception (description précise de différents aspects de la modélisation)

- ✓ *Principes fondamentaux: le type de concepts , théories ou principes généraux qui sous tendent la modélisation*
- ✓ *Emergence: quels caractères ou faits doivent être émergents des principes ou caractéristiques primaires de la modélisation → liés aux objectifs de la modélisation*
- ✓ *Adaptation. Les règles précisant les décisions individuelles et celles précisant l'évolution de ces règles*
- ✓ *Objectifs: quels sont les buts / objectifs suivies par les individus? Quels sont les mécanismes sélectifs et quelle mesure serait celle du succes d'un trait/ individus par rapport à un autre?*
- ✓ *Apprentissage: Comment les individus changent-ils leurs comportements en fonction de l'expérience accumulée ?*



## Pratique de bonne modélisation: le protocole The 'ODD' (Overview, Design concepts, and Details)

### □ Conception

- ✓ *Prediction.* Comment les individus estiment-ils les conséquences futures de leurs décisions présentes ? Comment estiment ils les conditions futures de leur environnement?
- ✓ *Sondage:* Quels états internes ou environnementaux sont-ils censés percevoir et considérer dans la prise de décision? Quels variables d'états (autres agents ou entités) sont ils capables de percevoir?
- ✓ *Interaction.* Quels types d'interaction entre agents? Interactions directes ou indirectes (à travers la compétition pour une même ressource)?
- ✓ *Stochasticité:* quels processus sont modélisés comme aléatoires ou partiellement aléatoires?

## Exemple : le dilemme du prisonnier modifié

### □ Pourquoi de la modélisation

- ✓ étudier la coopération dans un dilemme social
- ✓ Étudier les effets des structures d'interaction

### □ Décrire les Entités , variables, états , échelles

- ✓ Les individus essentiellement fixes
  - On fixe une contrainte : **pour la programmation en netlogo on utilisera que des patches pour nos individus (pas de déplacements nécessaire)**
- ✓ Variable des individus (patches):
  - my-agent-set ; a reference agent set for taking a decision and computing payoffs and wealth
  - decision ; the decision to cooperate or defect
  - emotion ; the emotion of defecting from a cooperative norm
  - prop-def ; % of defectors in the neighborhood
  - prop-coop ; % of cooperators in the neighborhood
  - my-wealth ; ; current gain of an individual from the prisoner's dilemma without emotion )
  - my-other-nearby ; agent with different action than me in my-agent-set

## Exemple : le dilemme du prisonnier modifié

### □ Décrire les Entités , variables, états , échelles

#### ✓ Variables globales :

- count-my-agent-set ; size of the interacting neighborhood
- coop-wealth ; average wealth of cooperators (gain from the prisoner's dilemma without emotion ) ;
- def-wealth; average wealth of defectors (gain from the prisoner's dilemma without emotion )
- %-coop-tot ; % of cooperator in the population
- segregation ; mesure of segregation in the population
- world-size ; population size
- Coop-rate ;initial cooperation rate ( curseur dans netlogo)
- Coop-cost ; cost of cooperation  $C$  ( curseur )
- nb-who-play-this-turn ; agents who change their current decision in the current period ( curseur dans netlogo)
- error-in-decision ; with a small probability the agent does not take an optimal decision ( curseur dans netlogo)

## Exemple : le dilemme du prisonnier modifié

### □ Processus , overview et étape:

#### ✓ Setup :

##### — Agents

- initialement un % (Coop-rate) d'agents sont coopérateurs (decision = 1) et un % (1- Coop-rate) sont défecteurs (decision = 0) .
- Les coopérateurs seront des patches verts et les défecteurs des patches rouges.
- Chaque agent a une variable “emotion” issue d'une uniforme [0 1]
- chaque agent à un voisinage de taille “count-my-agent-set”

##### — world-size : nombre total d'individus = 400

## Exemple : le dilemme du prisonnier modifié

### □ Processus , overview et étape: (cont)

- ✓ Procédure décrivant la dynamique du système « go » : trois procédures principales
  - **choose-strategy** ; only patches who play ( nb-who-play-this-turn ) update their decision ; other stick to current decision;
  - **compute-wealth** ; all patches update their wealth including those who did not change their decision
  - **update-globals** ; all globals variable (%-coop segregation are updated

## Exemple : le dilemme du prisonnier modifié

### □ Processus , overview et étape: (cont)

- ✓ **choose-strategy** ; chaque agent **appelé à jouer** ( nb-who-play-this-turn ) regarde la proportion d'agent coopérateur et défecteur dans son voisinage; il choisit la décision qui maximise le gain moyen
  - le gain moyen de la coopération est  $(1-C) \cdot \text{prop-coop} - C \cdot (1-\text{prop-coop})$ ;
  - le gain moyen de la défection est  $(1-X) \cdot \text{prop-coop}$
  - néanmoins, avec une probabilité “error-in-decision” il choisit une décision aléatoire

## Exemple : le dilemme du prisonnier modifié

### □ Processus , overview et étape: (cont)

- ✓ **compute-wealth** ; tous les agents réactualisent leur profit ( my-wealth) ;
  - Le profit ne tient pas compte des émotions mais est-ce que l'individu gagne effectivement : c'est les gain du dilemme du prisonnier (sans les émotions)

## Exemple : le dilemme du prisonnier modifié

### □ Processus , overview et étape: (cont)

#### ✓ update-globals ;

- on comptabilise le % de coopérateur dans la société pour la période considérée (%-coop-tot). On fait un graphique de %-coop-tot
- on calcule la ségregation et on fait un graphique de la segregation.
- La procédure de calcul de la ségregation est la suivante :
  - on calcule le nombre espéré de paires mixtes étant donné un pourcentage de coopérateur
    - soit  $k$  est la taille du voisinage et le taux de coopération (%-coop-tot)

nombre espéré de paires mixtes =  $k \times (\text{\%-coop-tot}) \times (1 - \text{\%-coop-tot}) \times \text{population size}$

- On calcule le nombre effectif de paires mixtes
  - Pour chaque agent : on comptabilise le nombre de joueurs avec une décision différente de cet agent dans le voisinage
  - On fait la somme de ce nombre sur tous les agents
  - On divise par deux pour avoir le nombre effectif de paires mixtes

**Segregation =  $1 - (\text{nombre effectif de paires mixtes} / \text{nombre espéré de paires mixtes})$**



## Exemple : le dilemme du prisonnier modifié

### □ Design concept (description précise de différents aspects de la modélisation)

- ✓ Emergence: coopération conditionnelle et ségrégation sociale
- ✓ Adaptation. Les agents prennent leur décision en maximisant le gain dans un dilemme du prisonnier modifié (avec une possibilité d'erreur dans le choix) . Contrainte sociale : la norme de la société est coopérer et cette norme s'applique à tous les individus, mais avec plus ou moins de force (valeur de la culpabilité à ne pas la suivre la norme)
- ✓ Objectifs: Maximiser leur gain dans un dilemme du prisonnier modifié
- ✓ Apprentissage adaptatif. Les agents estiment le taux de coopération future par le taux de coopération actuel dans leur voisinage. Soit: sur une seule période de temps et une taille de l'échantillon égale à la taille de leur voisinage.

## Exemple : le dilemme du prisonnier modifié

### □ Design concept

- ✓ *Prediction.* Les individus sont adaptatifs et font l'hypothèse que le taux futur de coopération est le même que l'actuel
- ✓ *Sondage (sensing):* chaque agent connaît son émotion et mémorise la décision actuelle de son voisinage
- ✓ *Interaction.* Localisée de tailles différentes . Pas de compétition pour une ressource
- ✓ *Stochasticité: la décision est aléatoire avec une faible probabilité*

## Travail à faire

1. Programmer ce modèle avec **des agents PATCH uniquement**
2. Faire une procédure calculant l'équilibre de Nash (dans le setup)
3. **Faire une analyse avec behavior space :**
  - ✓ il s'agira de produire sur un document word , de 2/3 pages maximum, des graphiques avec un commentaire reproduisant les principaux faits observés sur la DERNIERE PERIODE de la simulation ( → « mesure runs at every steps » de behaviorSpace est désactivé) . Les faits observés concernent:
    - explorer les **effets des structures d'interaction sur %-coop-tot et segregation**: (faites 25 (ou plus) répliquions pour chaque structure d'interaction .
    - count-my-agent-set = 4 , 8 , 24 , all agents (excepts me).
    - ON prendra le **cas coop-cost = 0.2**,
    - ; les autres paramètres sont fixés à : « nb-who-play-this-turn » = population size ; « error-in-decision » = 1% ; enfin 1000 périodes environ pour le nombre de ticks)
    - Il y a donc deux graphiques à produire : un pour **%-coop-tot** l'autre pour **segregation**

## Travail à faire

- faire varier coop-cost (choisir 5 à 10 valeurs ) et comparer l'équilibre de Nash avec l'équilibre moyen obtenu pour 25 replications dans le cas
  - count-my-agent-set = 4 (voisinage du type neighbors4)
  - “ tous sauf l'agent appelant “
  - Les autres paramètres : « nb-who-play-this-turn » = population size ; « error-in-decision » = 1% ; enfin 1000 périodes environ pour le nombre de ticks)
  - 1 seul graphique à produire
- Points bonus: explorer les effets des structures d'interaction sur la ségrégation pour une distribution sur les émotions avec un équilibre instable  $> 0.5$  et un équilibre stable  $<< 1$ .
- À déposer sur moodle pour le vendredi 23 janvier 12H. (attention les simulations avec behavior space prennent du temps).