
SCI - Dilemme itéré du prisonnier

Loïc Allart & Kévin Moulart

17 Dec 2013

Pré-requis

Pour compiler le projet, deux bibliothèques C/C++ doivent être installées au préalable.

Boost : <http://www.boost.org/users/download/>

Pour l'installer via apt-get : `apt-get install libboost-all-dev`

Python 3.3 : <http://www.python.org/download/releases/3.3.3/>

Compilation

Un Makefile est mis à disposition pour compiler sous Ubuntu. Les chemins indiqués pour Python et Boost sont les répertoires d'installation par défaut.

Deux tâches sont disponibles : « all » et « clean ».

Le Makefile génère un fichier exécutable nommé « ipd » (iterated prisoner's dilemma)

Utilisation

Le projet est livré avec un répertoire nommé « Strategies_pool ». Ce répertoire contient les stratégies codées en python.

L'exécutable prend en argument le chemin d'un répertoire contenant un sous-ensemble de ces stratégies. Exemple :

`ipd /fake/path/strategies`

Trois fichiers seront créés :

- `<date>_matrix.csv`, qui contiendra la matrice des scores.
- `<date>_evolution.csv`, qui contiendra la population de chaque stratégies au fil des itérations.
- `history.csv`, qui contient l'historique des tournois (une ligne par exécution du programme).

Pour spécifier la population d'une stratégie, il suffit de renommer le fichier qui l'implémente :

`all_c.py` → `200_all_c.py`

Par défaut, les stratégies ont une population de 100 individus.