Université de Caen Basse-Normandie
Département d'informatique
L2 informatique, 2018
Unité UE2

Sécurité et aide à la décision

Jeu d'infection

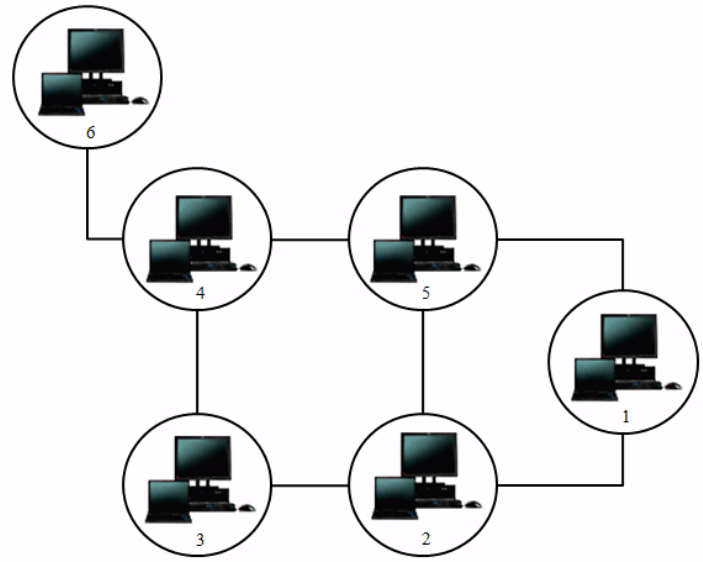
Grégory Bonnet

---

1 Jeu d'infection

Considérons le jeu suivant :

Soit un réseau de $M = \{1 \dots |M|\}$ machines tel que chacune soit reliée à un sous-ensemble de M, formant un graphe non-orienté comme illustré dans la figure ci-dessous.



Parmi ces machines, un sous-ensemble I est infecté. À chaque tour, l'attaquant peut infecter une machine de $M \setminus I$ reliée à une machine de I. À chaque tour, le défenseur peut supprimer une ou plusieurs arêtes associées à une machine de $M \setminus I$ (le défenseur ne peut pas supprimer un lien entre deux machines infectées).

Sachant que le défenseur est le premier joueur, le jeu s'arrête lorsque l'attaquant ne peut plus infecter de nouvelle machine. Le vainqueur est le joueur qui contrôle le plus de machine, sachant que le défenseur a pour objectif de minimiser le nombre d'arête à supprimer. En effet, son objectif est de maintenir son réseau le plus complet possible.

2 Implantation du jeu

Implémentez dans un langage de programmation vu en cours (Python ou Java) le jeu présenté ci-dessus. Vous implémenterez une méthode qui vous permet de générer aléatoirement un réseau et une infection initiale. Pour ce faire, utiliser un graphe aléatoire binomial : chaque arête entre deux machines existe avec une probabilité p.

3 Implantation des algorithmes de recherche

Écrivez dans un langage de programmation vu en cours (Python ou Java) une implémentation de (1) `minmax` ou `negamax` et (2) un élagage `alphabeta` (ou sa version `negamax`). Spécifiez à haut niveau (interfaces) les objets représentant le jeu et vous prendrez garde à bien obtenir le coup qui doit être joué. Considérez la fonction d'évaluation suivante :

$$f(s_i) = \sum_{m \in |M \setminus I|} \deg(m)$$

Implémentez un compteur pour obtenir le nombre de nœuds explorés au cours de la partie.

4 Intégration et expérimentations

Votre programme doit pouvoir être exécuté en lui passant six paramètres :

- le nombre total de machines
- le nombre de machines initialement infectées
- la probabilité que deux machines soient reliées
- la profondeur de raisonnement de l'attaquant
- la profondeur de raisonnement du défenseur
- l'utilisation ou non d'un élagage `alphabeta`

En considérant que la profondeur de raisonnement est la même pour les deux adversaires et en fixant le nombre de machines (infectées ou non), faites des expérimentations et tracez des graphiques représentant le nombre de nœuds explorés par `minmax` puis `alphabeta` selon (1) la profondeur de raisonnement, (2) la probabilité que deux machines soient reliées.

5 Travail demandé

Le TP devra être rendu pour le lundi 5 mars 13h30 au plus tard sur eCampus : `https://ecampus.unicaen.fr/mod/assign/view.php?id=88078`. Il s'agira d'une archive (ZIP, TAR, GZ au choix) contenant :

- votre code source (compilable ou exécutable sans les scories laissées par un IDE).
- un (éventuel) fichier `readme.txt` afin de d'indiquer la marche à suivre.
- un court rapport au format PDF présentant les résultats d'expérimentation.

Les noms des deux membres du binôme doivent être indiqués dans le rapport et le `readme.txt`. La notation reposera sur (1) les fonctionnalités développées : génération de jeu, `minmax` et `alphabeta` ; (2) les expérimentations réalisées ; (3) la propreté et la lisibilité du code ; (4) la pertinence des commentaires.