Projet IA : Trouver un coupe-cycles de sommets de taille minimale

- Le travail s'effectue en binôme ; un *unique* trinôme peut être accepté si le nombre d'étudiant.e.s est impair) ; sauf raison exceptionnelle justifiée aucun monôme n'est accepté.
- Votre travail, le code et un rapport, est à rendre avant le 30 juin à 23h59 sur Moodle.
- Votre code doit pouvoir être testé facilement et de manière intuitive. Il doit être bien documenté. Votre rapport ne se concentrera pas sur votre code mais plutôt sur la demarche pour implémenter et optimiser vos différentes méthodes de recherche et sur la présentation de vos résultats.
- Si vous utilisez des ressources (e.g., mémoires, papiers de recherche, autres cours, code), vous devez y faire référence dans votre rapport.

Description du problème. Dans un graphe orienté, $D = (V, A)$, un coupe-cycles de sommets est un ensemble $S \subseteq V$ de sommets tel que le graphe $D[V \setminus S]$ obtenu en enlevant les sommets de S du graphe est un graphe orienté acyclic. On s'intéresse au problème d'optimisation suivant : *Étant donné un graphe orienté $D = (V, A)$, trouver un plus petit ensemble S tel que S est un coupe-cycles de sommets.*

Travail demandé sur les méthodes de recherche. On désire résoudre ce problème par les méthodes de recherche locale vues au chapitre 3 du cours : méthodes hill-climbing, recuit simulé, algorithmes génétiques. On vous propose la formalisation suivante (mais vous êtes libres d'en proposer d'autres). Une solution sera un ensemble $T \subseteq V$; la fonction à minimiser sera le nombre de noeuds inclus dans au moins un cycle (un coupe-cycles a un score de 0) ; pour chercher un coupe-cycles de taille minimale vous pouvez utiliser vos méthodes en considérant uniquement des solutions de taille inférieure à une certaine limite qui évoluera au fur et à mesure de la recherche. Vous devrez également :

- formaliser la notion de voisinage entre les solutions ;

- formaliser les opérations de mutation et de cross-over (pour l'algorithme génétique) ;
- travailler sur différentes stratégies pour optimiser vos méthodes de recherche locale.

On vous conseille de réaliser le projet en Python en utilisant la bibliothèque *networkx* [1] qui permet de travailler facilement sur des graphes orientés et qui possède différentes méthodes reliées à la recherche de cycles dans des graphes.

Travail demandé sur le protocole expérimental. Vous devez proposer un protocole expérimental pour tester ces méthodes, mettre en place ces expérimentations, puis réaliser un rapport détaillant votre travail (e.g., présentation des stratégies d'optimisation et résultats expérimentaux). Vous devrez y détailler la mise-en-place et les résultats de vos expérimentations numériques : génération des instances, performances des différentes stratégies, temps de calcul des différentes stratégies et caractéristiques des ordinateurs utilisés...

Quelques remarques :

- La bibliothèque networkx vous permet d'utiliser certains modèles probabilistes de graphe, comme les graphes Barabási-Albert ou les graphes Erdös-Rényi. Vous pouvez utiliser ces modèles pour vos expérimentations.
- Une méthode standard pour mener des expérimentations numériques consiste à choisir plusieurs paramètres des instances générées. On fera ensuite varier chacun de ces paramètres, *toutes choses égales par ailleur*, afin de tester l'impact de ces paramètres sur certaines mesures (e.g., performance ou temps de calcul).
- Si vous utilisez un modèle probabiliste de graphe, alors pour chaque jeu de paramètres, il ne faudra pas générer qu'une seule instance de graphe mais plutôt un grand nombre d'instances sur lesquels vous pourrez calculer une moyenne et un écart type.

References

[1] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.