R5.11 – Méthodes d'optimisation pour l'aide à la décision

Guilherme Dias da Fonseca

TP 1

1 Ensemble Indépendant ou Stable

Un insemble indépendant ou stable est un sous-ensemble des sommets d'un graphe tel que aucune arête relie deux sommets du sous-ensemble.

Entrée : Un graphe non-dirigé $G = (V, E)$, où V est l'ensemble de sommets et E est l'ensemble d'arêtes.

Sortie : Un sous-ensemble indépendant $I \subseteq V$. On dit que I est *indépendant* s'il n'y a pas d'arête entre deux sommets de I.

Objectif : Maximiser la cardinalité $|I|$.

1.1 Exemple

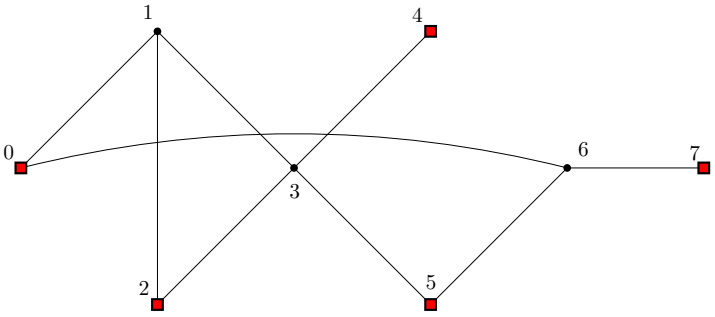


FIGURE 1 – Exemple de graphe et son stable de taille $|I| = 5$.

Disons que la rentrée est le graphe avec l'ensemble de sommets

$$V = \{0, 1, 2, 3, 4, 5, 6, 7\}$$

et l'ensemble d'arêtes

$$V = \{\{0, 1\}, \{0, 6\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{3, 4\}, \{3, 5\}, \{5, 6\}, \{6, 7\}\}.$$

Un ensemble indépendant de taille $|I| = 5$ est $I = \{0, 2, 4, 5, 7\}$. Voir figure 1.

1.2 Applications

Les ensembles indépendants maximaux, dans le domaine des mathématiques et de l'informatique, ont des applications variées dans plusieurs domaines. Voici quelques-unes de ces applications :

- Algorithmes de graphe : Les ensembles indépendants maximaux sont utilisés dans la théorie des graphes pour résoudre des problèmes tels que la coloration de graphes, la recherche de cliques maximales (l'ensemble indépendant maximal du graphe complémentaire), la planification des horaires, etc.
- Réseaux de communication : En télécommunications, les ensembles indépendants maximaux peuvent être utilisés pour minimiser les interférences et maximiser l'efficacité du spectre radioélectrique.
- Bioinformatique : Dans l'analyse des données génétiques et des réseaux biologiques, les ensembles indépendants maximaux peuvent être utilisés pour identifier des ensembles de gènes ou de protéines interagissant de manière cohérente.
- Analyse de réseaux sociaux : Dans les réseaux sociaux en ligne, les ensembles indépendants maximaux peuvent aider à identifier des groupes d'utilisateurs selon leurs interactions ou intérêts.
- Le problème de packing fait référence à une classe de problèmes d'optimisation combinatoire qui impliquent le placement d'objets dans des conteneurs de manière efficace, en respectant certaines contraintes ou objectifs spécifiques. Une fois qu'on trouve un ensemble finit d'emplacements possibles, le problème de maximizer le nombre d'objets dans un containeur peut être modélisé comme trouver un stable maximal.

2 Logiciel

Vous allez écrire un logiciel qui prend comme entrée un fichier qui encode le graphe et produit comme sortie un autre fichier avec la liste des sommets du stable. En générale, vous n'allez pas trouver le stable le plus gros possible (personne sait le faire !), mais on cherche quand même des gros ensembles.

Vous devez écrire le logiciel avec le langage C++ et vous pouvez vous servir du code qu'on a développé en Développement Efficace (mon code aussi, en donnant le crédit dans les commentaires). Votre code ne doit pas avoir d'interface graphique ni aucune interaction avec l'utilisateur et je dois pouvoir compiler et exécuter votre code sur Linux. Pour des raisons pratiques, le temps d'exécution de votre logiciel doit être limité à 5 minutes par fichier d'entrée.

Je vous rappelle que copier ou partager le code est interdit et risque des sanctions très sévères. Je vous prévois que je vais passer votre code dans un détecteur de plagiat de code C++ avant la correction.

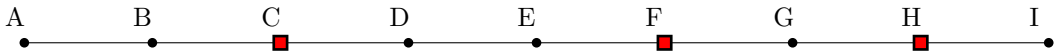
2.1 Algorithme glouton

D'abord, on va utiliser une méthode gloutonne (greedy) où à chaque pas on choisi un sommet à mettre dans la solution et ensuite on supprime ce sommet ainsi que ces voisins du graphe. On répète jusqu'à avoir vidé le graphe.

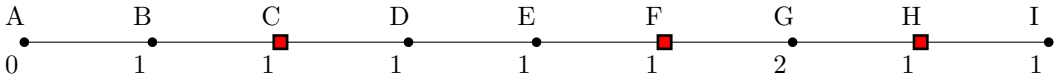
Le critère pour bien choisir un sommet à chaque pas est très important. Il faut penser qu'on préfère un sommet qui a peu de voisins. Ajoutez des facteurs aléatoires pour essayer plusieurs solutions possibles et garder la meilleure solution trouvée.

2.2 Recherche locale

Ensuite, si vous avez le temps, vous pouvez améliorer la solution gloutonne en utilisant la recherche locale. Il s'agit d'une technique qui modifie localement une solution de façon à potentiellement l' améliorer. On va illustrer la technique avec un graphe très simple et un stable de taille 3.

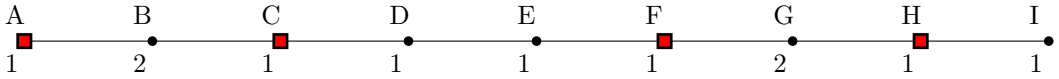


On peut sauvegarder le nombre de voisins de chaque sommet qui sont dans le stable. Ici, on utilise le voisinage fermé, c'est à dire, le sommet soi même est inclus dans son voisinage.

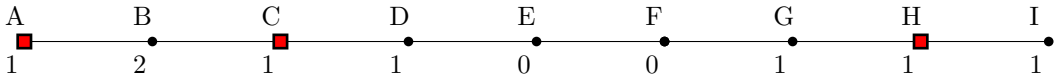


On va faire les opérations suivantes dans un boucle.

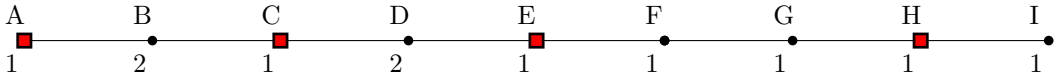
D'abord, tant qu'il y a des sommets avec numéro 0, on en choisi un au hasard (A, le seul dans ce cas) et on l'ajoute au stable. On met à jour les numéros des voisins.



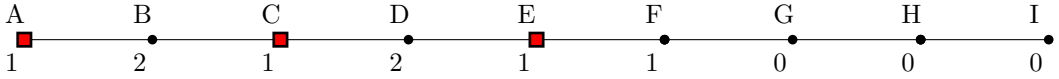
Ensuite, on choisi un sommet du stable au hasard (F) et on l'enlève du stable. On met à jour les numéros de ses voisins.



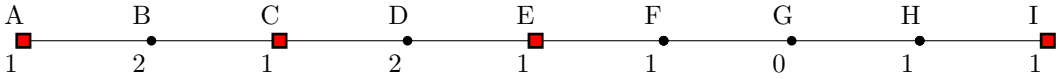
On répète la boucle. Disons qu'on ajoute E cette fois.



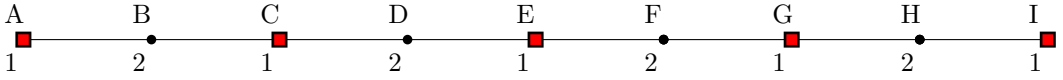
Et on supprime H.



On répète la boucle et ajoute I.



Et G.



On peut continuer à répéter la boucle plusieurs fois jusqu'à depacer un temps fixé. Évidement on veut renvoyer le stable avant et pas après en supprimer un sommet.

2.3 Fichiers d'entrée

Le fichier d'entrée `.edges` est textuel et chaque contient deux entiers séparés par un espace, qui indiquent qu'il y a une arête entre les deux sommets. Chaque arête apparait seulement une fois dans le fichier, soit dans une direction, soit dans l'autre. Noté qu'avec ce format, on ne peut pas représenter des sommets sans arête.

Pour l'exemple on a le fichier :

```
0 1
0 6
1 2
1 3
2 3
3 4
3 5
5 6
6 7
```

Il y a 9 fichiers d'entrée sur l'ENT, ayant de 30 à 40000 sommets et de 109 à 119960 arêtes. À titre de curiosité, ces fichiers ont été créés de 3 façons différentes, toujours à partir des ensembles de points du plan que j'ai trouvé en ligne. Dans les fichiers `udg` (*Unit Disk Graph*), je relie tous les pairs de points dont la distance est inférieur à une valeur r choisie. Dans les fichiers k`nn` (*k-Nearest Neighbor*), je relie chaque point aux k points les plus proches de lui. Dans les fichiers `del`, j'utilise la *triangulation de Delaunay*.

Notez qu'un graphe n'a pas obligatoirement des coordonnées x et y pour les sommets. Par exemple, un graphe d'amitié sur Facebook n'a pas de coordonnées. Pour ce TP, j'ai choisi de n'utiliser que des graphes basés sur les coordonnées du plan de façon à rendre le résultat plus visuel. Vous avez le droit d'utiliser ces coordonnées si vous voulez, mais c'est pas obligatoire. À savoir, les coordonnées sont stockées une dans les 5 derniers chifres et l'autre dans les autres chiffres de l'indice du sommet.

J'ai aussi mis à disposition les fichiers svg avec une visualisation des graphes où la souris sur le sommet affiche le numéro du sommet.

2.4 Fichier de sortie

Le format du fichier de sortie est simple. On liste sommets de I, un par ligne. Le fichier a $|I|$ lignes avec un entier par ligne. Pour l'exemple où $I = \{0, 2, 4, 5, 7\}$ on a le fichier :

```
0
2
4
5
7
```

On utilise l'extension `.ind` pour ces fichiers.

Je vais mettre sur Ametice un outil en python `tester.py` capable de vérifier vos solutions ainsi que de produire un fichier SVG pour que vous puissiez les visualiser.

3 Rendre le TP

Le TP sera noté et la qualité des solutions trouvées va influencer sur la note. Vous devez rendre un fichier zip avec :

1. Le code source avec votre nom dans la première ligne
2. Un script bash appelé `build.sh` pour compiler votre code sur Linux et exécuter votre logiciel avec tous les fichiers d'entrée en produisant automatiquement les fichiers de sortie.
3. Les fichiers de sortie avec le même nom des fichiers d'entrée correspondants mais l'extension `.ind` à la place de `.edges`.

Pensez que j'ai beaucoup de TPs à corriger, alors pour que je puisse rendre cette tache plus efficace je vais coder des logiciels pour faire quelques parties automatiquement. Pour ça, il faut avoir exactement les bons noms des fichiers (minuscule et majuscule c'est pas pareil). Parfois, je vois des trucs très bizarres comme des fichiers rar renommé zip pour qu'Ametice l'accepte et bien sur que le script que j'écris pour décompresser les fichiers ne marche pas ! Le non respect des consignes peut être pénalisé.