

Graphes et optimisation combinatoire

Projet

Danial, Ryan, Guillaume, Laurent

Faculté des Sciences , section Info
Université de Mons



December 9, 2019

1 Introduction

■ Introduction

2 GRASP

■ Greedy Randomized Construction

■ Local Search

■ Algorithme principal

3 Résultat

4 Conclusion

5 Source

- Nous avons choisit de coder notre projet en Java.
- Nous avons utiliser GRASP comme métaheuristique.

Greedy Randomized Construction

On commence avec une solution vide et un certain nombre de candidats (les sommets). On va ensuite évaluer le coût d'ajout de chaque candidat. Grace à cela, nous allons créer une liste de candidat restreint où intervient le paramètre α qui nous permettra de varier entre la diversification et intensification. On choisit un des sommets (au hasard) dans la liste que l'on ajoute à la solution. Cet algorithme se finira s'il n'y a plus de candidat.

Local Search

■ Mouvement

Nous avons d'abord du créer un voisinage grâce à un mouvement (on prend un ensemble au hasard dans la solution ensuite nous allons prendre chaque sommet de cet ensemble que nous allons intégrer dans les autres ensembles afin générer un nouveau voisinage (en tenant compte de la faisabilité).

Parmi les voisinages, nous allons prendre une solution meilleure.
L'algorithme s'arrête quand il n'y a plus de meilleure solutions.

Algorithme principal

L'algorithme va itérer un certain nombre de fois (donné en paramètre). Nous commençons avec une solutions nulle et à chaque itération, on utilise le Greedy Randomized Construction. Si cette solution est possible, nous faisons une recherche locale qui renvoie possiblement une meilleure solution et si cette solution est meilleure que notre solution actuelle on remplace notre solution par la nouvelle.

Résultat

Fichier	Arêtes	Sommets	α	Modularité
Fichier 1	28	16	0.5	0.6071428571428571
Fichier 2	50	20	0.5	0.5480000000000007
Fichier 3	65	25	0.5	0.5692307692307693
Fichier 4	160	80	0.5	0.7375
Fichier 5	613	115	1	0.6011419249592123

Conclusion

La métaheuristique que nous avons utilisé convient bien au problème et nous donne des résultats satisfaisants.

Source

- GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURES
- Community Detection by Modularity Maximization using GRASP with Path Relinking