Partition en sous graphes denses : complexité et modélisation LocalSolver

Julien Darlay

LocalSolver
24 avenue Hoche, 75008 Paris, France
jdarlay@localsolver.com

Mots-clés : *partition de graphes, complexité, heuristique*

La détection de communautés est un problème d'analyse de données où les informations sont représentées sous forme de graphe. Les observations forment les sommets du graphe et les arêtes modélisent une interaction entre deux observations. Par exemple, un graphe de collaborations scientifiques peut être construit en considérant qu'un sommet est un chercheur et qu'une arête relie deux individus ayant une publication commune.

En pratique, ces graphes d'interactions se divisent en communautés contrairement à des graphes aléatoires. De manière intuitive, une communauté est un ensemble d'individus ayant de fortes interactions entre eux et peu d'interactions avec l'extérieur. De nombreux modèles ont été proposés dans la littérature [4].

Dans ce travail, nous présentons un modèle basé sur la densité définie Goldberg [3]. Nous donnons des résultats de complexité et d'approximabilité sur les problèmes liés à l'optimisation de ce critère. En pratique, nous montrons comment obtenir de bonnes solutions pour ce problème à l'aide de LocalSolver.

1 Formulation mathématique

Soit $G = (V, E)$ un graphe simple, sa densité $d(G)$ est définie par le ratio : $d(G) = \frac{|E|}{|V|}$. Le problème classique lié à cette définition de la densité consiste à trouver le sous-graphe $G[X]$ induit par un sous-ensemble des sommets X de densité maximum. Ce problème peut être résolu en temps polynomial en utilisant la programmation linéaire [1] ou par des algorithmes de flots [3]. Lorsque le nombre de sommets du sous-graphe est fixe, le problème est difficile [1].

Soit Π l'ensemble des partitions de V sans classe vide. Nous définissons la *densité* $d(P)$ d'une partition $P \in \Pi$ par la formule suivante :

$$d(P) = \sum_{X \in P} d(G[X]) = \sum_{X \in P} \frac{|E(X)|}{|X|}$$

Le problème de partition qui nous intéresse ici est de trouver la partition P qui maximise la densité $d(P)$. Chaque classe de la partition correspond alors à une communauté.

2 Complexité et heuristique de résolution

Le problème de décision associé à la recherche de la partition de plus grande densité est NP-complet en effectuant une réduction depuis le problème classique de coloration de graphe [2]. Cette réduction préserve l'approximabilité du problème et par conséquent il a le même facteur d'approximation que le problème de coloration. Lorsque le graphe G est un arbre, la partition de plus grande densité peut être trouvée en temps polynomial par un algorithme de type programmation dynamique [2].

```
//Variable de décisions, x[i][c] = 1 si le sommet i est dans la classe j
x[1..n][1..n] <- bool();

//Affectation de chaque sommet à une classe
for [i in 1..n]  constraint sum[c in 1..n] (x[i][c]) == 1;

//Nombre de sommets et d'arêtes par classe
card[c in 1..n] <- max(1,sum[i in 1..n] (x[i][c]));
edges[c in 1..n] <- sum[i in 1..m](x[origin[i]][c] * x[dest[i]][c]);

//Objectif
maximize sum[c in 1..n] (edges[c] / card[c]);
```

FIG. 1 – Modèle LocalSolver pour la partition en graphes denses.

La résolution pratique du problème n'est pas aisée avec les outils de programmation linéaire en nombres entiers, en particulier à cause de la présence de plusieurs ratios et de termes quadratiques. Nous avons choisi une approche par recherche locale avec le logiciel LocalSolver [1]. Ce solveur de programmation mathématique est de type model & run, c'est-à-dire qu'il prend en entrée une modélisation dans un formalisme mathématique simple et la résolution est ensuite totalement automatique. Par exemple, dans le cas du partitionnement en graphes denses, le modèle peut s'écrire en quelques lignes (voir Figure 1).

Les résultats présentés portent sur différents types de graphes : les arbres où la solution optimale est connue, des graphes aléatoires pour lesquels la structure de la solution optimale est connue ainsi que des instances de la littérature. Sur les arbres, les solutions trouvées sont optimales ou quasi-optimales (moins de 2% d'écart sur des arbres à 1000 sommets). Pour les graphes dont la structure de communautés est connue, les résultats montrent un biais du critère utilisé : les solutions comportent plus de classes que le résultat attendu. On observe le même biais sur les instances de la littérature. En fixant le nombre de classes dans la solution, on retrouve des solutions proches des partitions attendues.

Références

[1] M. Charikar. Greedy approximation algorithms for finding dense components in a graph *APPROX*, 2000

[2] J. Darlay, N. Brauner et J. Moncel. Dense and sparse graph partition *Discrete Applied Mathematics*, 160, 2012

[3] A. Goldberg. Finding a Maximum Density Subgraph *EECS Department, University of California, Berkeley*, 1984

[4] M. Newman et M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69, 2004.

1. www.localsolver.com