Problème d'attribution de fréquence
Maximisation de couverture

Yohann CHARREIRE--KIRBACH - 4aIR SI - Stage réalisé au LAAS-CNRS, Toulouse et encadré par Emmanuel HEBRARD et Laurent HOUSSIN

Description du problème

Un problème d'attribution de fréquence (FAP) désigne le fait de chercher à attribuer une fréquence, ou une bande de fréquences, aux différents utilisateurs d'un réseau sans fils tout en tenant compte de paramètres tels que le nombre limité de fréquences disponibles ou les interférences entre utilisateurs.

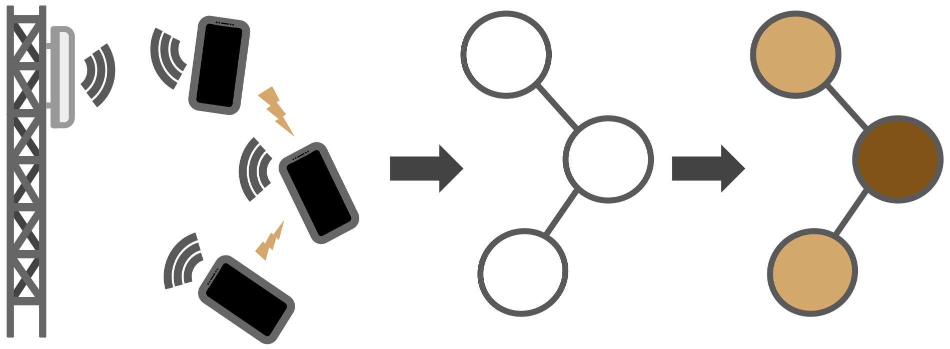

Fig.1 : Résolution du problème d'attribution de fréquence pour un système simple à trois utilisateurs.

Ce type de problème peut-être ramené à la coloration du graphe qui représente les interférences au sein du système. Un utilisateur est alors un noeud, un risque d'interférence est une arête et une fréquence assignée est une couleur.

Il existe de nombreuses versions de ce problème, mais lors de ce stage c'est celle de couverture maximum à k couleurs qui fut abordée. C'est-à-dire qu'on cherche à couvrir le plus d'utilisateurs possibles, sans interférence, en étant limité à k fréquences. Le problème étant NP-difficile, l'objectif est d'obtenir la couverture maximum le plus rapidement possible.

Approche naïve

La première approche fut de résoudre le problème en programmation linéaire en nombre entier (PLNE) en utilisant la bibliothèque Python de Gurobi. Il faut alors définir nos variables de décision.

Si le noeud i est de couleur k alors $x_{i,k}$ vaut 1 sinon il vaut 0. On peut alors énoncer notre objectif (0) et la première contrainte (1) : "Un noeud n'a qu'une couleur, ou aucune". En notant E, l'ensemble des arêtes, on peut énoncer la seconde contrainte (2): "Deux noeuds adjacents n'ont pas la même couleur".

$$Obj. : max \sum_{\forall k} \sum_{\forall i} x_{i,k} \tag{0}$$

$$s.t. : \forall i, \sum_{\forall k} x_{i,k} \leq 1 \tag{1}$$

$$\forall (i,j) \in E, \forall k, x_{i,k} + x_{j,k} \leq 1 \tag{2}$$

Fig.2 : Représentation du problème de couverture maximale sous forme canonique

Bien que cette approche fonctionne, le temps de résolution était bien trop grand même pour des graphes de taille raisonnable.

Génération de colonnes

La seconde approche fut d'avoir recours à la génération de colonne décrite par Malaguti, Monaci, et Toth, en utilisant Cplex en C++. Cette méthode consiste à résoudre un problème défini sur un ensemble exponentiel de variables (colonnes), mais en ne générant qu'une partie de cet ensemble.

Nous avons ici envisagé le problème de coloration minimum, c'est à dire une couverture complète en cherchant à utiliser le moins de couleurs possible.

Dans ce cas, les colonnes choisies sont des stables maximaux, c'est-à-dire un ensemble de noeuds qui ne sont pas voisins et auxquels on ne peut pas rajouter de noeud. Ainsi si on attribue une couleur à une colonne, on l'attribue à tous les noeuds de celle-ci. Le problème peut donc se voir comme une couverture par ensembles, où les ensembles correspondent aux colonnes..

Nous avons tenté de générer des colonnes par PLNE et en ayant recours à des algorithmes de construction de stables maximaux . Cependant, bien que rapide, ces derniers pourrait être modifiés pour donner des stables améliorant mieux notre solution.

Bilan

Au terme de ce stage, des erreurs subsistent dans le code, ce qui mène à de mauvais résultats pour certains types de graphe. De plus d'autres algorithmes plus performant pour la construction de colonne pourraient être envisagé. Cependant le code produit lors de stage est un premier pas dans l'implémentation de l'approche de Malaguti, Monaci, et Toth au sein du framework d'Emmanuel Hebrard et George Katsirelos. Cela permettra à terme d'hybrider les deux méthodes. On pourra alors espérer profiter du calcul de bornes fortes de la génération de colonne et de l'apprentissage de clauses afin de renforcer le modèle pendant la recherche.