# Partitionnement d'un ensemble connexe d'hypergraphes orientés sans cycle avec minimisation de longueur de chemins

Julien Rodriguez[1,2], François Galea[1], François Pellegrini[2], Lilia Zaourar[1]

[1]Université Paris-Saclay, CEA, List
F-91120, Palaiseau
prenom.nom@cea.fr

[2]Université de Bordeaux, INRIA
francois.pellegrini@u-bordeaux.fr
julien.rodriguez@inria.fr

**Mots-clés** : *hypergraphe, partitionnement, chemin critique.*

## 1 Introduction au partitionnement d'hypergraphes

Un hypergraphe est une extension de la notion de graphe dans lequel les hyperarêtes contiennent au moins un sommet. Les hypergraphes modélisent de nombreux objets comme par exemple, un circuit électronique, un ensemble de données ou un ensemble de tâches/calculs à réaliser.

Le partitionnement d'un hypergraphe consiste à séparer les sommets en plusieurs sous-ensembles appelés parties, de taille plus ou moins similaire. L'objectif est de minimiser le nombre d'hyperarêtes partagées entre les parties. Il existe plusieurs façons de mesurer la valeur d'une partition, comme le nombre d'hyperarêtes coupées ou la connectivité$-1$. Notons $y_{jk}$ la variable qui détermine si l'hyperarête $j$ appartient à la partie $k$, alors il est possible de définir la connectivité$-1$ pour une hyperarête $j$ par : $\sum_k y_{jk} - 1$.

Le partitionnement d'hypergraphes qui consiste à minimiser ces objectifs est un problème difficile [4]. Dans ce travail, nous nous intéressons au problème de partitionnement sur une sous-classe d'hypergraphes modélisant des circuits électroniques. Ces hypergraphes sont composés de sous-hypergraphes orientés sans cycles, inter-connectés par les sommets sources et les sommets puits de ces derniers. Ci-dessous, la Figure 1 expose un exemple dans lequel les sommets blancs sont les sommets sources/puits interconnectant les sous-hypergraphes entre eux.

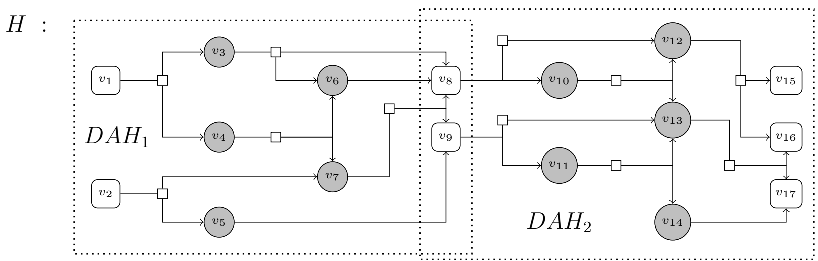


FIG. 1 – Exemple d'un hypergraphe composé de deux sous-hypergraphes orientés sans cycle

Nous proposons aussi d'étendre la fonction objectif mesurant le coût de coupe afin tenir compte de la minimisation de la longueur des chemins au sein des sous-hypergraphes orientés sans cycle. Dans notre contexte, une hyperarête coupée sera pénalisée et augmentera la longueur des chemins incluant cette dernière. À partir d'une partition $x$ de $H = (V, E)$, d'une fonction de poids $d_i$ associé au sommet $i$ et d'une variable $z_{jp}$ indiquant si l'hyperarête $j$ est coupée dans $p$, il est possible de définir $p_x$ le coût du plus long chemin dans $H$ :

$$p_x = \max_{h \in H} \max_{p \in P_h} \sum_{i \in p} d_i + \sum_{j \in p} z_{jp}$$

Notons que quand il n'y a qu'une seule partie ($k = 1$), alors le calcul de $p_x$ fournit une borne inférieure $\underline{p}$ pour $k > 1$. Remarquons aussi, que le calcul du plus long chemin au sein des sous-hypergraphes sans cycle se calcule efficacement en utilisant l'ordre topologique.

## 2 Modélisation du problème et approches de résolution

Soit un hypergraphe $H = (V, E)$, avec $x_{ik}$ qui vaut 1 si le sommet $v_i$ est placé dans la partie $k$, 0 sinon. Alors le modèle dans [2] étendu à la minimisation de chemin, s'exprime comme :

$$\min\ A \times \sum_j w_j \left(\sum_k y_{jk} - 1\right) + B \times \max_{h \in H} \max_{l \in P_h} \left(\sum_i p_{li} d_i + \left(\sum_j z_{jl}\right) \times D\right)$$

$$\begin{aligned}
st.\ \sum_k x_{ik} &= 1, \forall i \\
h_{ij} x_{ik} &\leq y_{jk}, \forall i, j, k \\
h_{ij} h_{i'j} p_{li} p_{li'} (x_{ik} - x_{i'k}) &\leq z_{jl}, \forall i, i', j, k, l \\
h_{ij} h_{i'j} p_{li} p_{li'} (x_{i'k} - x_{ik}) &\leq z_{jl}, \forall i, i', j, k, l \\
\sum_i q_{ir} x_{ik} &\leq c_{kr} \times (1 + \epsilon), \forall k, r \\
x_{ik}, y_{jk}, z_{jl} &\in \{0, 1\}, \forall, i, j, k, l
\end{aligned}$$

Paramètres :

- $h_{ij} \ = \ 1$ ssi $v_i \in E_j$
- $c_{kr}$ : quantité $r$ max pour la partie $k$
- $q_{ir}$ : quantité $r$ requise par $v_i$
- $d_i$ : délai associé au sommet $v_i$
- $p_{li} \ = \ 1$ ssi $v_i \in p_l$
- $D$ : la pénalité de coupe

où $A$ et $B$ sont deux paramètres permettant d'ajuster l'importance d'un objectif par rapport à l'autre. Il est également possible de faire de la scalarisation et de placer l'objectif $p_x$ en une contrainte forte : $p_x \leq \overline{p}$, en le faisant évoluer de façon dichotomique avec :

$$\overline{p} \leq \max_{h \in H} \max_{p \in P_h} \sum_{i \in p} d_i + \sum_{j \in p} D$$

Dans la littérature, une majorité de méthodes de partitionnement sont des heuristiques multi-niveaux [3] constituées de trois étapes, illustrées dans la Figure 2.

La première étape consiste à réduire la taille de l'instance en appliquant à plusieurs reprises un algorithme de clustering. Cette étape prend fin lorsque la taille de l'hypergraphe réduit est suffisamment petite, permettant ainsi de passer à la deuxième étape qui à pour but de calculer une partition de $H$, dite *partition initiale*. Enfin, la troisième et dernière étape consiste à raffiner la solution en remontant tous les niveaux de clustering en appliquant à chaque niveau une heuristique qui essaye d'améliorer la solution courante par le biais de déplacement des sommets entre parties.

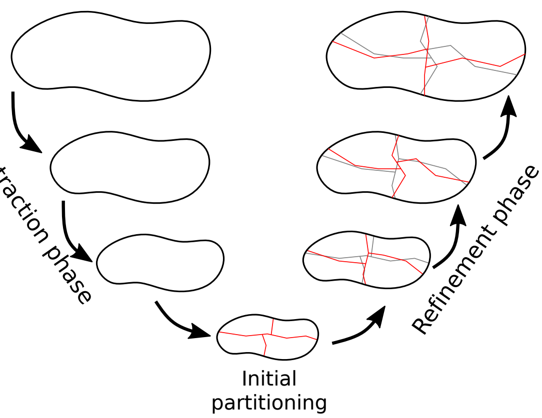


FIG. 2 – Partitionnement multi-niveaux

Dans notre étude, nous proposons de comparer deux méthodes. La première consiste à partir d'une solution basée sur l'ordre topologique donnant un bon candidat pour l'objectif $p_x$ et de l'améliorer avec un algorithme K-FM [1] afin minimiser la taille de la coupe. La seconde méthode consiste à partir d'une solution initiale intéressante du point de vue de la taille de coupe et d'utiliser l'algorithme de K-FM dans une version étendue pour minimiser $p_x$.

## Références

[1] C.M. Fiduccia and R.M. Mattheyses. A linear-time heuristic for improving network partitions. In *19th Design Automation Conference*, pages 175–181, 1982.

[2] Lilia Zaourar François Galea. Partitionnement multi-contraint d'hypergraphes valués avec sommets pré-fixés. 2020.

[3] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning : applications in vlsi domain. 7(1) :69–79, 1999.

[4] Thomas Lengauer. Combinatorial algorithms for integrated circuit layout (chap. 6). In *Applicable theory in computer science*, 1990.