An approximation algorithm for hypergraph disjoint clustering problem with path-length awareness

Julien Rodriguez, François Galea, François Pellegrini, Lilia Zaourar

An Approximation Algorithm for Hypergraph Disjoint Clustering Problem with Path-length awareness

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

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

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

Mots-clés : *hypergraph, clustering, critical path.*

1 Introduction

Circuit partitioning is a usual process in Very Large-Scale Integrated (VLSI) design. Indeed, many classical methods (placement of logic cells on the silicon surface or resource mapping on a FPGA) do not scale well with ever-increasing circuit sizes. The typical size often vary from millions to billions of gates, making such methods challenging. Therefore, a possible approach is to cluster the circuit to reduce its apparent size for critical processing operations, while trying to achieve a good level of locality within the clusters. Even for combinatorial circuits, there are several models for the clustering problem. In particular, we consider here the problem of clustering without replication in combinatorial circuits. Our main objective is to minimize the overall delay. Recently, this problem has been studied only from an algorithmic and complexity point of view [1]. We extend the problem to cluster a set of connected Directed Acyclic Hypergraphs (DAH) [2] and propose a dedicated approximation algorithm.

2 Problem statement

Let $\mathcal{H} \overset{\text{def}}{=} (\mathcal{V}, \mathcal{A}, \mathcal{W}_v, \mathcal{W}_a)$ be defined as a set of vertices $\mathcal{V}$ and a set of hyperarcs $\mathcal{A}$, with a vertex weight function $\mathcal{W}_v : \mathcal{V} \to \mathbb{N}$ and a hyperarc weight function $\mathcal{W}_a : \mathcal{V} \to \mathbb{N}$ where each hyperarc $a \in \mathcal{A}$ is a subset of vertex set $\mathcal{V} : a \subseteq \mathcal{V}, \forall a \in \mathcal{A}$. Let us define $\mathcal{V}^R$ the red vertex set, and $\mathcal{V}^B$ the black vertex set, such that $\mathcal{V}^R \cap \mathcal{V}^B = \emptyset$ and $\mathcal{V}^R \cup \mathcal{V}^B = \mathcal{V}$. $\mathcal{H}$ is a Directed Acyclic Hypergraph (DAH) whose source and sink vertices are in $\mathcal{V}^R$, and its other vertices are in $\mathcal{V}^B$. Since a hypergraph can contain multiple DAHs in the general case, it is possible to represent this set of DAHs by a red-black hypergraph. Let $\mathbf{H} \overset{\text{def}}{=} \{\mathcal{H}_i, i \in \{1, n\}\}$ be a set of DAHs in which every $\mathcal{H}_i$ is a DAH. An example can be found in Fig. 1. Let us define $\mathcal{P}$ as the set of red-red paths in $\mathbf{H}$, such that $\mathcal{P} \overset{\text{def}}{=} \{p | p \text{ is a path in } \mathcal{H}, \forall \mathcal{H} \in \mathbf{H}\}$. From these paths and a function $d_{max}(u, v)$ which computes the maximum distance between vertices u and v of some DAH $\mathcal{H}$, we can now define the longest path distance for $\mathcal{H}$ as : $d_{max}(\mathcal{H}) \overset{\text{def}}{=} max(d_{max}(u, v) | u, v \in \mathcal{H})$, and, by extension, for $\mathbf{H}$ as : $d_{max}(\mathbf{H}) \overset{\text{def}}{=} max(d_{max}(\mathcal{H}) | \mathcal{H} \in \mathbf{H})$.

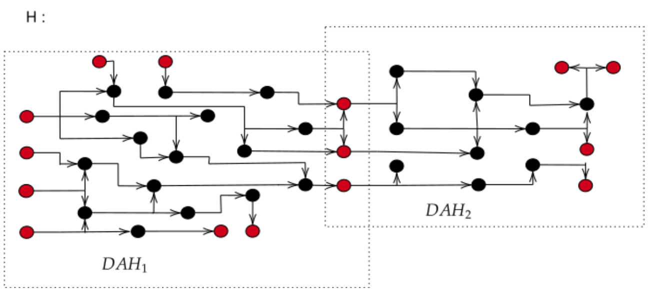

FIG. 1 – Hypergraph with 2 DAHs

A cluster is a subset of vertices such that each path length passing through different clusters is penalized by a constant D, that is, if u and v are not in the same cluster : $d_{max}(u,v) \geq d_{max}(u,v) + D$. The objective function f_p can therefore be defined as the minimization of the longest path of $\mathbf{H}$ subject to clustering C : $f_p = \min d_{max}(\mathbf{H}^C)$. The problem $CN{<}\mathcal{V}, M, \Delta{>}$ consists in finding a clustering of the vertices such that M bounds the sizes of the clusters and the cost function f_p is minimized. This problem is known to be NP-Hard for $D \geq 1$ [1].

3 Solving methods

The classical Heavy-Edge Matching (HEM) algorithm can be used to cluster a vertex set. In this algorithm, vertices connected with heaviest arcs are merged prioritarily. Weighting schemes can significantly impact solution quality, and should be tailored to the problem to solve. A possibility to compute arc weights is to propagate or back-propagate the path length, so that arc weights represent the length of the longest path passing through the arc. We present a weighting scheme in which each arc weight is the length of the local critical path passing through this arc.

HEM can be applied recursively to produce quickly clusters of a maximum prescribed size $M \geq 2$. However, recursive clustering may bring sub-optimal solutions compared to direct k-way clustering, as illustrated in Fig. 2.

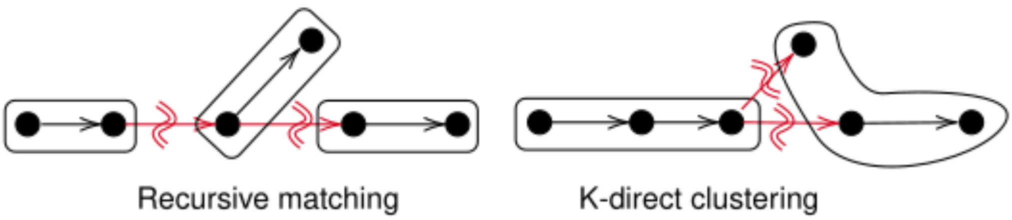

FIG. 2 – Example of clustering with $M = 3$. Recursive matching yields three clusters with two delays on the critical path and cannot coarsen more, while direct 3-clustering yields two clusters with only one delay.

Let us define w_j as the weight of the arc j. As described above, for $a_j = (u,v)$, $w_j = d_{max}(u,v)$. Let f_p^{obj} a target value for a clustering result. We can define a cutting capacity as : $cut\_cap(a_j) = \frac{f_p^{obj} - d_{max}(u,v)}{D}$.

The principle of the algorithm is to find a minimum value of $f_p^{obj} \in [d_{max}(\mathbf{H}),\ d_{max}(\mathbf{H}) \times D]$, by binary search, while respecting the cluster size constraint. The sizes of the clusters are directly constrained by the cutting capacity (cut_cap) of arcs defined by f_p^{obj}. Each arc with a cutting capacity equal to zero will be packed in the same cluster.

4 Conclusions and perspectives

To sum-up, in this study, we propose to compare several weighting schemes and approaches, including one based on the recursive HEM and our approximation algorithm. We show that our algorithm and is an $M - OPT$ approximation algorithm in its parameterized version and performs in $O(m.log(m))$, with m the number of hyperarcs. We plan to include this algorithm in a multi-level scheme for red-black hypergraph partitioning with path-length minimization [2].

Références

[1] Z. Donovan, K. Subramani, and V. Mkrtchyan. Analyzing clustering and partitioning problems in selected vlsi models. *Theory of Computing Systems*, 64, 10 2020.

[2] J. Rodriguez, F. Galea, F. Pellegrini, and L. Zaourar. Partitionnement d'un ensemble connexe d'hypergraphes orientés sans cycle avec minimisation de chemin. In *23ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision*, Villeurbanne - Lyon, France, Feb. 2022. INSA Lyon.