SERIE B — INFORMATIK

Modeling Hypergraphs by Graphs with the same Mincut Properties ◇

Edmund Ihler*
Dorothea Wagner**
Frank Wagner***‡

B 92–22
October 1992

Abstract

An elegant and general way to apply graph partitioning algorithms to hypergraphs would be to model hypergraphs by graphs and apply the graph algorithms to these models. Of course such models have to simulate the given hypergraphs with respect to their cut properties.
An edge-weighted graph (V, E) is a *cut-model* for an edge-weighted hypergraph (V, H) if the weight of the edges cut by any bipartition of V in the graph is the same as the weight of the hyperedges cut by the same bipartition in the hypergraph. We show that there is no cut-model in general.
Next we examine whether the addition of dummy vertices helps: An edge-weighted graph $(V \cup D, E)$ is a *mincut-model* for an edge-weighted hypergraph (V, H) if the weight of the hyperedges cut by a bipartition of the hypergraphs vertices is the same as the weight of a minimum cut separating the two parts in the graph. We construct such models using positive *and* negative weights. On the other hand, we show that there is no mincut-model in general if only positive weights are allowed.

◇This work was done while the authors were with THE LEONARDO FIBONACCI INSTITUTE for the Foundations of Computer Science.
*Theoretische Informatik, Eidgenössische Technische Hochschule Zürich, CH-8092 Zürich, e-mail: ihler@inf.ethz.ch.
**Fachbereich Mathematik, Technische Universität Berlin, Straße des 17. Juni 136, W-1000 Berlin 12, Germany, e-mail: dorothea@combi.math.tu-berlin.de.
***Institut für Informatik, Fachbereich Mathematik, Freie Universität Berlin, Arnimallee 2-6, W-1000 Berlin 33, Germany, e-mail: wagner@tcs.fu-berlin.de.
‡This work was partly supported by the ESPRIT Basic Research Action, Projekt ALCOM II.

1 Introduction

The layout of circuits on chips or printed circuit boards is typically done via a divide-and-conquer approach. Usually the circuit is described as a list of small circuit elements and a netlist where each net consists of a set of elements that have to be connected electrically (cf. [5], [6]). As this is mathematically just a hypergraph, the divide-step can be described as the problem to find a cut, i.e. to partition the vertex set of a hypergraph, with minimum cut weight.

A lot of the existing partitioning algorithms are either developed only for ordinary graphs (e.g. [1]) or work much faster on them (e.g. [7]). The bifurcator layout concept of [2] is developed only for ordinary graphs, too.

An elegant and general way to apply graph partitioning algorithms to hypergraphs would be to model hypergraphs by graphs and apply the graph algorithms to these models as suggested in [5, chapter 6.1.5]. Of course such models have to simulate the given hypergraphs with respect to their cut properties.

We are given hypergraphs with positive weights on their hyperedges. We are looking for graphs modeling them, which in the most general case may have additional vertices and positive and negative weights on their edges. For technical reasons we allow zero weight edges and always assume the model graphs to be complete. We also discuss restrictions to the original vertex set or to non-negative weights.

Since we are not interested in modeling just a certain specific hypergraph but in a method to model hypergraphs in general it suffices to examine the existence of models for hypergraphs consisting of just a single unit weight hyperedge: If there are such models we can construct a model for an arbitrary positive weighted hypergraph by multiplying the edge weights by appropriate factors and putting together the models for all the hyperedges. On the other hand, if there is no model even for a single hyperedge there is definitely no such model in general.

So in the rest of the paper we will use the terms cut-model and mincut-model in the following sense:

Definition 1 *A graph (V, E) on k vertices is a* cut-model *(for a unit weight hyperedge on k vertices) if the weight of any cut induced by a non-empty proper subset W of V is equal to one.*

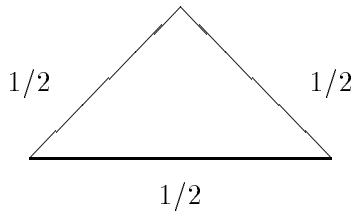
A graph $(V \cup D, E)$ on $k+d$ vertices is called a mincut-model *(for a unit weight hyperedge on k vertices) if for every non-empty subset W of V we have that the weight of any cut with minimum weight (*mincut*) under those separating W from $V \setminus W$ is equal to one. It must be zero for $W = \emptyset$.*

A *cut* is formally a bipartition of the vertex set V. As discussed in [6] the appropriate notion of the *cut weight* is the sum of the weights of those (hyper-)edges that have vertices in both parts of the partition. By $w(A)$ we denote this sum where A is one of the parts. The weight of the single edge from x to y is denoted as $w(\overline{xy})$. If A and B are two disjoint subsets of V, $w(A, B)$ denotes the sum of the weights of those (hyper-)edges that have vertices in A and B. So, $w(A) = w(A, V \setminus A)$.

2 Cut-Models

We start with the only positive fact concerning cut-models: Figure 1 shows a cut-model for three vertices.

Figure 1:



Beside that we have:

Theorem 2 *There is no cut-model for $k \geq 4$.*

Proof:

Suppose there is such a model and let x and y be two vertices of it, then we have

$$w(\{x\}) + w(\{y\}) = w(\{x, y\}) + 2w(\overline{xy}).$$

On the other hand

$$w(\{x\}) = w(\{y\}) = w(\{x, y\}) = 1.$$

So, since x and y are chosen arbitrarily, $w(e) = 1/2$ for all $e \in E$.

Obviously such a uniformly weighted clique is far from having constant cut weight for $k \geq 4$.

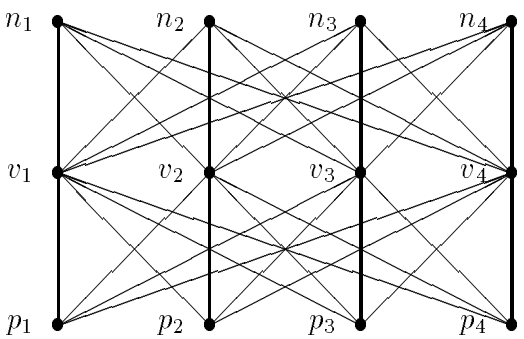
□

3 Mincut-Models

To model a hyperedge on k vertices $v_1, \ldots, v_k$, we add two sets of dummy vertices $n_1, \ldots, n_k$ and $p_1, \ldots, p_k$, and build a complete tripartite graph on the three sets (cf. Figure 2 ($k = 4$)).

Edges from v_i to n_i have the large negative weight N, edges from v_i to p_i have the large positive weight P. All other edges have a small (positive) weight p. Intuitively, thus every mincut is forced to cut every edge weighted by N and to avoid cutting any of the edges weighted by P. This induces already the two parts of the mincut. Then the number of other edges cut is always the same, namely half of them are cut, since an edge $\overline{v_i n_j}$ with $i \neq j$ is cut if and only if $\overline{v_i p_j}$ is not.

Figure 2:



So, every such cut has weight $kN + k(k-1)p$. It remains to choose the weights such that they add up to one, that the described cuts are really minimum, and that no mincut having all the v_i's on one side has negative weight.

Choosing $p = 1/k$ and $N = -(k-2)/k$ gives the desired sum; together with $P = 1$ this guarantees the two other properties:

Let w.l.o.g. the two sets to be separated by a mincut be $v_1, \ldots, v_i$ and $v_{i+1}, \ldots, v_k$ for an $i \in \{1, \ldots, k-1\}$ and assume, that the dummy vertices are not divided as described above. Then simple calculations show that in every case the switch of a "misplaced" dummy vertex to the other part decreases the cut weight by at least p.

Finally, cutting off any dummy vertex from all the v_i's cuts edges of weight at least p so putting all the vertices together yields the minimum weight "cut" in this case.

An important ingredient of this construction is the use of negative weighted edges. That this is unavoidable is shown in the following:

Theorem 3 *There is no mincut-model for $k \geq 4$ if negative weights are not allowed.*

Proof:

We show the theorem for $k = 4$. For larger k we can partition the set V of non-dummy vertices of the assumed mincut-model arbitrarily into four non-empty parts. This yields a model for $k = 4$ by identification of all non-dummy vertices of each part.

Suppose there is a mincut-model for $k = 4$. Let $V = \{s, t, x, y\}$ and let A and B be mincuts separating $\{s, x\}$ from $\{t, y\}$, and $\{s, y\}$ from $\{t, x\}$, respectively.

We will use a general property of such "crossing mincuts" which is well known (cf. e.g. [3], page 48/49). We sketch its proof for completeness:

Lemma 4 *Let A and B be two mincuts separating a vertex s from a vertex t, then $A \cap B$ and $A \cup B$ are such s-t-mincuts, too, and $w(A \setminus B, B \setminus A) = 0$.*

Proof:

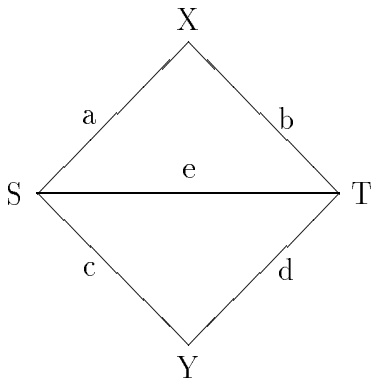
$$w(A) + w(B) = w(A \cap B) + w(A \cup B) + 2w(A \setminus B, B \setminus A) \geq w(A) + w(B)$$

The equality holds since on the left and right side the same edges are cut. The inequality holds since both $A \cap B$ and $A \cup B$ are cuts separating s from t, and all weights are positive.

□

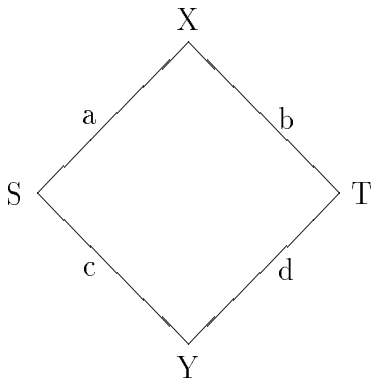
The two mincuts A and B partition $V \cup D$ into four subsets containing one non-dummy vertex each, which we call S, X, Y and T. If we identify all vertices (dummy or not) in each of the four parts we get a condensed graph (cf. Figure 3), where a, b, c, d, and e are the sums of the weights of the edges connecting the parts. Because of the lemma there is no edge connecting X and Y.

Figure 3:



Next, observe that A and $(V \cup D) \setminus B$ are x-y-mincuts both, so there is one edge less in the condensed graph (cf. Figure 4)

Figure 4:



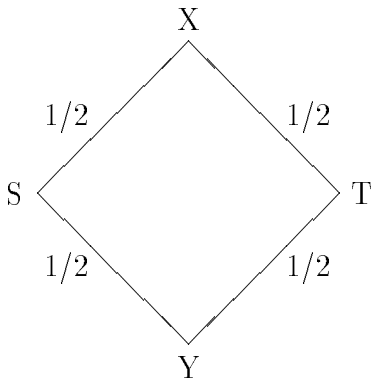
By the lemma again we know, that S is an s-t-mincut and that X is an x-y-mincut. All of the cuts discussed so far have by assumption weight one, so

$$b + c = a + d = a + c = a + b$$

yielding

$$a = b = c = d = 1/2$$

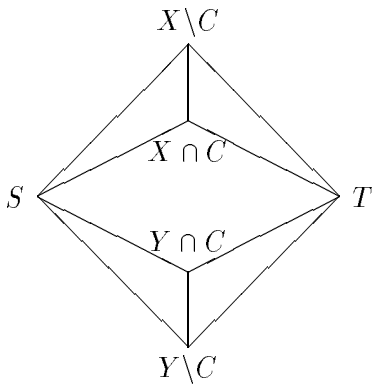
Figure 5:



As the last instrument of this proof, we now construct a mincut C separating $\{x, y\}$ and $\{s, t\}$ with the additional property that $C \subseteq X \cup Y$. We start with an arbitrary mincut C'' separating $\{x, y\}$ and $\{s, t\}$. Since C'' and $(V \cup D) \setminus A$ are x-s-mincuts, $C' = C'' \cap ((V \cup D) \setminus A)$ has weight one, too. But then C' is also an x-t-mincut. The same holds for $A \cup B$ and thus for $C = C' \cap (A \cup B)$. With these two intersections we have shaved off the parts of C'' contained in S and T respectively, yielding a C with the desired properties.

If now, in our condensed graph, we split X into $X \setminus C$ and $X \cap C$, and Y into $Y \setminus C$ and $Y \cap C$ we have the situation depicted in Figure 6.

Figure 6:



Now, since $X \cap C$ and $Y \cap C$ separate x and y, respectively, from all the other real vertices their cut weights are at least one, so:

$$w(C) = w(X \cap C) + w(Y \cap C) \geq 2.$$

That, of course is a contradiction to the minimality of C. □

4 Approximate Mincut-Models—An Open Problem

As we have seen, there is no positive weighted mincut-model for a hypergraph, in general. But, what can we expect below this perfect, but non-existing simulation? We look instead for an approximate mincut-model for unit weight hyperedges, that is as balanced as possible, which means that the quotient between the largest and the smallest weight mincut partitioning the set of real vertices is as small as possible. This is captured in the following definition:

Definition 5 *A positive weighted graph $(V \cup D, E)$ on $k+d$ vertices is called* approximate mincut-model *(for a unit weight hyperedge on k vertices) if the minimum weight of any mincut separating a proper and non-empty subset W of V from $V \setminus W$ is equal to one. The weight of the largest such mincut is called the* balance *of the model.*

Lengauer [5] conjectures that cliques without any dummy vertices are the best balanced models:

Conjecture 6 *There is no approximate mincut-model having a smaller balance than the uniformly weighted complete graph on k vertices; i.e. every edge has weight $1/k-1$ and the balance is slightly more than $k/4$.*

Observe that e.g. the balance of a binary tree with unit edge weights having the real vertices as its leafs is roughly twice as bad.

From [4] and [7] we know that no approximate cut-model (without dummy vertices) has a better balance than the clique.

References

[1] E. R. Barnes. *An algorithm for partitioning the nodes of a graph*, SIAM Journal on Algebraic and Discrete Methods **3** (1982) 541-550

[2] S. N. Bhatt, F. T. Leighton. *A framework for solving VLSI graph layout problems*, Journal of Computer and System Sciences **28** (1984) 300-343

[3] A. Frank. *Packing Paths, Circuits, and Cuts—A Survey*, in: B. Korte et al. (eds.). *Paths, Flows, and VLSI-Layout*, Springer, Berlin (1990) 47-100

[4] S. W. Hadley. *A New Approximation Technique for Hypergraph Partitioning Problems*, manuscript, University of Waterloo (1990)

[5] T. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout*, Teubner, Stuttgart, Wiley, Chichester (1990)

[6] D. G. Schweikert, B. W. Kernighan. *A Proper Model for the Partitioning of Electrical Circuits*, Proceedings of the Ninth ACM/IEEE Design Automation Workshop (1972) 57-62

[7] A. Vannelli, S. W. Hadley. *A Gomory-Hu Cut Tree Representation of a Netlist Partitioning Problem*, IEEE Transactions On Circuit and Systems **37** (1990) 1133-1139