Pioneer Research on Mathematical Models and Methods for Physical Design

Chris Chu
Department of Electrical and Computer Engineering
Iowa State University
Ames, Iowa
cnchu@iastate.edu

ABSTRACT
In the inaugural International Symposium on Physical Design (ISPD) at 1997, Prof. Te Chiang Hu has delivered the keynote address “Physical Design: Mathematical Models and Methods” [1]. Without any question, Prof. Hu has made a lot of foundational and profound contributions to physical design automation and to computer science and mathematics in general. This paper highlights several of Prof. Hu’s pioneer works related to flow and cut in a flow network to commemorate his achievements.

CCS CONCEPTS
- Mathematics of computing → Network flows; - Theory of computation → Network flows; - Hardware → Physical design (EDA).

KEYWORDS
Physical design automation; Network flow; Cut

ACM Reference Format:

1 INTRODUCTION
Finding a flow or a cut with specific property in a flow network has a lot of applications in diverse fields. In VLSI design, a circuit can be modeled as a network. A maximum flow in the network characterizes the connectivity between two components in the circuit. A minimum cut provides a partitioning of the circuit with the least dependency between the two partitions. In addition, many optimization problems in VLSI design can be transformed into either a flow or a cut problem. For example, the Lagrangian multiplier update problem in a Lagrangian relaxation based gate sizing algorithm is formulated as a minimum cost flow problem [2]. The layout decomposition problem in double patterning lithography is reduced to a maximum cut problem in a flipping graph [3].

 Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:

2 MULTI-COMMODITY NETWORK FLOWS
The multi-commodity flow problem has many practical applications, e.g., modeling of messages in a communication network, different goods in a transportation system, and traffic in a road network. In VLSI design, routing in circuits can be modeled as a flow in a network. To handle the routing of multiple nets, we can use a multi-commodity flow model in which each of the nets is represented by one commodity. Multi-commodity flow based approaches have been applied to various formulations of VLSI routing problems (e.g., [4–8]).

Prof. Hu is one of the earliest researchers who works on the multi-commodity flow problem. He presented the seminal paper [9] which generalizes the max-flow min-cut theorem of Ford and Fulkerson [10] to the problem of finding the maximum simultaneous flows of two commodities.

Consider a connected network with positive arc capacities such that the arc capacity from node $N_i$ to $N_j$ is the same as that from node $N_j$ to $N_i$. Suppose $k$th kind of flow is from node $N_k$ to node $N_{k'}$ and is denoted by $F(k; k')$. Let $f(k; k')$ denote the value of $F(k; k')$. Let $c(k; k')$ denote the capacity of the minimum cut separating node $N_k$ and node $N_{k'}$. As a generalization of cut, let a disconnecting set for $k$ pairs of nodes $N_i, N_{i'}$ be a set of arcs, the removal of which will disconnect $N_i$ from $N_{i'}$ $(i = 1, 2, \ldots, k)$, and no proper subset of which will have this property. The value of a disconnecting set is the sum of capacities of the arcs in the set.

**Theorem 1.** (Max Bi-Flows Min-Cut Theorem) Two flows $F(1; 1')$ and $F(2; 2')$ are feasible if and only if (1), (2), (3) below are
all satisfied:

\[
\begin{align*}
    f(1; 1') & \leq c(1; 1') \\
    f(2; 2') & \leq c(2; 2') \\
    f(1; 1') + f(2; 2') & \leq c(1, 2; 1', 2')
\end{align*}
\]

The maximum sum of the two flows is equal to the minimum-cut capacity of all cuts separating the two pairs of nodes; i.e.,

\[
\text{max}[f(1; 1') + f(2; 2')] = \text{min}[c(1 - 2; 1' - 2'), c(1 - 2'; 1' - 2)].
\]

To prove this theorem, Prof. Hu has presented an algorithm similar to the labeling method for finding maximum flow of a single commodity to construct the two flows. The max-flow min-cut theorem is later extended to multicommodity flows by Onaga [11] and Iri [12].

3 MAXIMUM CONCURRENT FLOWS AND MINIMUM CUTS

In VLSI physical design and other applications, we often need to find the minimum cost cut separating a given pair of nodes. In [13], Prof. Hu together with Prof. Cheng have generalized the problem to finding all minimum cost cuts which separate all \( \binom{n}{2} \) pairs of nodes. They showed that for arbitrary costs (e.g., usual cut [10], weighted sparsest cut [14], or flux cut [15]), there are only \( n - 1 \) essential minimum cuts out of all \( 2^{n-1} - 1 \) possible cuts.

\textbf{Theorem 2.} Given a network with \( n \) nodes and an arbitrary cut cost function, we need at most \( n - 1 \) distinct cuts, such that for all pair of nodes, one of the \( n - 1 \) cuts is the minimum cut separating the pair.

They have also presented an algorithm to find the set of essential cuts with only \( n - 1 \) calls to an oracle which generates the minimum cut for a given node pair with respect to a given cost function.

Among the distinct cuts in the essential cut set, we may find the global minimum cut which is the cut with minimum cost among all \( 2^{n-1} - 1 \) possible cuts. In [13], Prof. Hu and Prof. Cheng focused on the ratio cut cost function, which is also called the weighted sparsest cut [14]. The problem of finding the global minimum ratio cut is NP-hard [16]. They proposed an approach by leveraging the relationship between global minimum ratio cut and the maximum concurrent flow [17]. The maximum concurrent flow problem, which maximizes the uniform flow demand between every pair of nodes, can be formulated as a linear programming problem and solved using column-generating techniques [18]. The saturated arcs in the maximum concurrent flow define a \( K \)-way partition of the network. Their key contribution is showing that if \( K \leq 4 \), then there exists a two-way partition of the partitioned \( K \) subsets which is the global minimum ratio cut.

4 A REPLICATION CUT FOR TWO-WAY PARTITIONING

In VLSI design, when a circuit is partitioned, it is often beneficial to allow some cells to be replicated. For example, when a large circuit is implemented by several FPGAs, the limited pin count of FPGA chips and the significant delay and power overhead for off-chip communications are often the bottleneck. By replicating some cells into multiple FPGAs, the demand in pin count and off-chip communications can be reduced. The effect of replication in reducing interchip connections is illustrated in Figure 1.

In [19], Prof. Hu and his collaborators have investigated the problem of two-way min-cut partitioning with cell replication. They first considered networks with only two-pin nets and without constraints on partition size. Given two nodes \( s \) and \( t \) to be separated, they introduced a novel replication graph such that an optimal replication partition can be constructed from the maximum flow in the replication graph. The replication graph is derived by first formulating the replication partitioning problem as a linear program and next interpreting its dual linear program as a network flow problem. The replication graph corresponding to the network flow can then be constructed. The structure of the replication graph is illustrated in Figure 2 and an example is shown in Figure 3.

**Figure 1:** Effect of replication. (a) The min-cut has a cut cost of 13 without replication. (b) Replicating \( R \) results in a cut cost of 4. [19]

**Figure 2:** Structure of replication graph \( G^* \). [19]

From Figure 2, we can see that the replication graph \( G^* \) basically consists of a copy of the original graph \( G \) and another copy \( G' \) similar to \( G \) but with all arcs reversed. Each node in \( G \) is connected to its corresponding nodes in \( G' \) with an arc with infinite capacity. A super source node \( s' \) (a super sink node \( t' \)) connecting to the source nodes (from the sink nodes) in \( G \) and \( G' \) with infinite capacity arcs is also added.

The optimum replication cut of \( G \) with respect to node pair \( s \) and \( t \) can be found by a maximum-flow minimum-cut solution of \( G^* \) with respect to node pair \( s' \) and \( t' \). Suppose the maximum-flow minimum-cut solution partitions the nodes of \( G \) into \( X \) and \( \bar{X} \) and the nodes of \( G' \) into \( X' \) and \( \bar{X'} \) as illustrated in Figure 2. Let \( S = X \), \( T = \{ij' \in \bar{X'}\} \) and \( R = V - S - T \). Then the optimum solution is to replicate \( R \) such that the two subsets are \( S \cup R \) and \( T \cup R \).

To handle VLSI applications, the idea of replication graph is extended to release the requirement of separating two given nodes,
to allow multiple-pin nets, and to enforce size constraints on partitions. Then the FM algorithm is extended to minimize a directed cut cost under size constraints. The extended FM algorithm is applied to the proposed replication graph to find a minimum-cost replication cut.

The presented algorithms are both elegant and useful in practice that their contribution is recognized by the 1997 IEEE Circuit and System Society Best Paper Award.

5 OPTIMAL LINEAR ORDERING
A fundamental problem in VLSI placement is the optimal linear placement problem, in which the gates of a circuit are placed along a line with minimum total wirelength. A special version of optimal linear placement is optimal linear ordering in which a weighted graph is placed in uniformly spaced slots. The optimal linear ordering problem is useful for placement in chips with regular layout fabrics like FPGA and gate array as well as for non-VLSI applications. Unfortunately, it is NP-complete [20].

In the seminal paper [21], Prof. Hu and his collaborator have presented two interesting results on optimal linear ordering. First, for an arbitrary graph, based on non-trivial relationship between optimal linear ordering and network flow, they established a lower bound on the total wirelength.

**Theorem 3.** For an arbitrary graph, the total cut capacity of the \( n - 1 \) fundamental cuts constructed by the Gomory-Hu algorithm [22] is a lower bound on the total wirelength of the optimal linear ordering.

Second, they considered another case in which the graph is a rooted tree. The rooted tree imposes a partial ordering on the nodes. A node \( x \) should precedes a node \( y \) in the linear order if \( x \) is an ancestor of \( y \) in the rooted tree. For a rooted tree, they presented an algorithm which requires \( O(n \log n) \) operations. They also showed the equivalence of the optimal linear ordering problem for a rooted tree to a job sequencing problem solved by Horn [23].

6 THE ORIENTATION OF MODULES BASED ON GRAPH DECOMPOSITION
After the placement of a VLSI circuit, the modules can be flipped to reduce wirelength and improve routability. This is a very practical problem and many heuristic algorithms have been proposed, e.g., analytical method [24], neural network approach [25], simulated annealing approach [26], simple greedy heuristics [27–30], linear programming / mixed integer linear programming based heuristics [31], and path-based optimization methodology [32]. An optimal symbolic algorithm based on Boolean Decision Diagram (BDD) has also been proposed but it can only be used for small size circuits as it is very slow [33].

Prof. Hu and his collaborators are among the earliest who have worked on the flipping problem [34]. They assumed that multi-pin nets have already been decomposed into two-pin nets. They have made several fundamental contributions.

First, they showed that the flipping problem can be transformed into the minimum cut problem of a graph with positive and negative capacities. Given a circuit with \( n \) modules, they constructed a graph with \( n + 1 \) nodes: \( n \) nodes represent the \( n \) modules which may be flipped, and a supernode \( T \). The graph has an interesting property that for any cut, the cut value is equal to the change in the total wirelength if all nodes on the same side as \( T \) are unflipped and all those on the other side are flipped. Consequently, the minimum cut implies a flipping solution with minimum wirelength. To achieve this property, consider a net \( s \) connecting two modules \( u \) and \( v \). Let \( C1 \) be the change in wirelength when only \( v \) is flipped, \( C2 \) be the change in wirelength when only \( u \) is flipped, \( C3 \) be the change in wirelength when both \( u \) and \( v \) are flipped. A triangle graph is devised as illustrated in Figure 4. The arc capacities \( c_{uT}, c_{vT}, \) and \( c_{uv} \) are uniquely determined by three simultaneous equations below:

\[
\begin{align*}
    c_{uT} + c_{uv} &= C1 \\
    c_{uT} + c_{uv} &= C2 \\
    c_{uT} + c_{vT} &= C3.
\end{align*}
\]

It is clear that for any cut of the triangle graph, the cut value always equals to the wirelength change of \( s \) in the corresponding flipping solution. To construct the graph for the whole circuit, we just superimpose the triangle graphs of all nets.

Second, they also proved that the flipping problem is NP-complete by reducing the minimum cut problem of a graph with positive and negative capacities, which is NP-complete [20], to the flipping problem.

Third, to handle large circuits in practical applications, techniques were presented to decompose the graph into subgraphs and
Figure 4: A net s connecting modules u and v and its triangle graph. [34]

to condense the nodes to speed up the search for minimum cut without sacrificing optimality.

ACKNOWLEDGEMENT
I would like to thank Prof. C. K. Cheng and Prof. A. B. Kahng for their invaluable suggestions to this paper.

REFERENCES