When Clusters Meet Partitions: New Density-Based Methods for Circuit Decomposition

Dennis J.-H. Huang and Andrew B. Kahng

UCLA Computer Science Department, Los Angeles, CA 90024-1596 USA
jenhsin@cs.ucla.edu, abk@cs.ucla.edu

Abstract

Top-down partitioning has focused on minimum cut or ratio cut objectives, while bottom-up clustering has focused on density-based objectives. In seeking a more unified perspective, we propose a new sum of densities *measure for multi-way circuit decomposition, where the* density *of a subhypergraph is the ratio of the number of edges to the number of nodes in the subhypergraph. Finding a k-way partition that maximizes the sum of k subhypergraph densities is NP-hard, but an efficient flow-based method can find the* optimal *(maximum-density) subhypergraph in a given hypergraph. Based on this method, we develop a heuristic which in practice has less than 10% error from an optimal sum of densities decomposition. Other results suggest that density-based heuristics can capture cut-based objectives, whereas the converse would seem difficult.*

1 Introduction

A fundamental task in VLSI layout applications is finding a "natural" decomposition of the netlist hypergraph into k disjoint subsets. Two types of objectives have been used.

First, *minimum cut* objectives minimize the total number of signal nets crossing between different components, with possible size constraints on the components; the Fiduccia-Mattheyses (FM) heuristic [4] is a leading example. More recent research has focused on *ratio cut* bipartitioning [14], which minimizes $\frac{C(P_1,P_2)}{|P_1|\cdot|P_2|}$ where $C(P_1, P_2)$ is the number of signal nets crossing between the two partitions P_1 and P_2. This metric is "natural" in that it addresses both the minimum cut and the partition size balance objectives.

Second, *density-based* decompositions intuitively view a *cluster* as a densely connected group of circuit elements. The goal is to identify "good" clusters, typically in bottom-up fashion. To assess whether a cluster is good, several clustering metrics have been proposed, e.g., Cong and Smith [3] defined a measure of cluster density as $\frac{|E|}{C(n,2)}$, where $|E|$ is the number of edges and n is the number of nodes in the cluster. This metric is biased to small clusters since $C(n, 2)$ increases quadratically in n (a cluster with one edge and two nodes (i.e., a 2-clique) would have high density). Many clustering objectives are difficult to evaluate, let alone optimize.

The main theme of our work is that (cut-based) partitioning and (density-based) clustering have no obvious common ground for "middle" values of $k \equiv$ number of components in the circuit decomposition. Yet, due to their common application to hierarchical circuit layout, these approaches arguably should "meet in the middle" (e.g., in mapping a system design onto a multiple-FPGA prototyping architecture, or in high-level synthesis applications).

Thus, we propose a new *maximum sum of densities* objective for k-way circuit decomposition. Formally, the *density* of a hypergraph $H = (V, E)$ is the ratio of the number of hyperedges to the number of nodes, i.e., $\frac{|E|}{|V|}$. Given a node subset $V' \subset V$, the density of the sub-hypergraph $H' = (V', E')$ induced by V' is the number of hyperedges in E that are *completely* contained in E', divided by the number of nodes in V'.

The main problem that we address is:

The k-way Maximum Sum of Densities (k-MSD) Problem: Given a hypergraph $H(V, E)$, divide V into k partitions, $P_1, P_2, \ldots, P_k$, such that the sum of the densities of the induced subhypergraphs is maximum.

In addressing the k-MSD problem, we will also discuss a more basic problem:

The Maximum Density Subhypergraph (MDS) Problem: Given a hypergraph $H(V, E)$, find a subhypergraph of H with largest density.

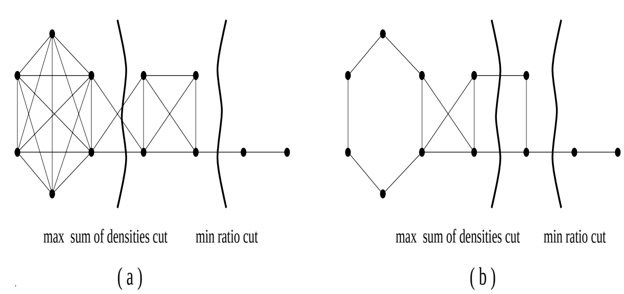

Figure 1: (a) The minimum ratio cut gives an uneven partitioning, while the maximum sum of densities cut gives a more balanced and natural partitioning. (b) When the circuit structure is changed, the maximum sum of densities cut shifts accordingly, while the minimum ratio cut remains the same.

Our hope is that the k-MSD objective can lead to "natural" circuit decompositions, i.e., our work is similar in spirit to that of Wei and Cheng [14], who in 1989 proposed ratio cut as a "more natural" decomposition objective (five years later, a large fraction of partitioning research now addresses some form of ratio cut). There are easy examples for which ratio cut does *not* give a "natural" circuit bipartition, while maximizing the sum of the two partition densities gives a more natural bipartition. Figure 1 shows that the ratio cut objective can overemphasize the reduction in net cut, while ignoring the internal structure of the partitions. The maximum sum of densities cut will shift as the internal structure of the graph changes, while the minimum ratio cut remains the same.

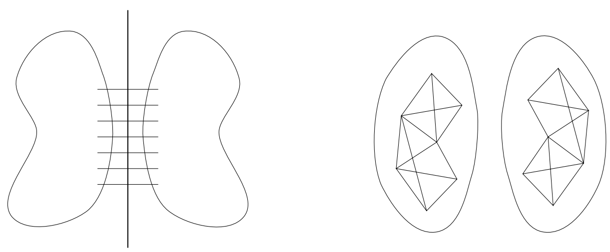

Figure 2: (a) Cut objectives pertain to the cutsize at the cluster boundary, while (b) density objectives pertain to the implied area or wirelength inside the cluster.

A secondary motivation is suggested in Figure 2: the cut objective has an *external* view of the circuit components, while the density objective has an *internal* view. Cutsize at the boundary of a component has been correlated with area and wirelength via the Rent parameter analysis, but the number of terminals affects area and wirelength only in a probabilistic sense, since shorter than expected routes may be possible. Density measures may yield less trivial lower bounds on layout area and wirelength. A final motivation, noted in Section 4 below, is that maximizing the sum of densities *can* capture both cut-based and density-based objectives, unlike any cut-based formulation.[1]

2 An Optimal Maximum Density Subhypergraph Solution

Many researchers have solved the maximum density subhypergraph problem when the input is a *graph* [11, 6]; we use *MDSG* to denote this special case. The usual transformation is to a series of minimum cut computations, which are accomplished using maximum flow techniques. Picard and Queyranne [11] formulated MDSG as a special case of 0-1 fractional programming, and used $O(n)$ flow computations to solve the problem. Goldberg [6] developed a different solution which requires only $O(\log n)$ flow computations.

[1] There are also superficial similarities between the k-MSD and certain cut-based objectives. If the given graph is a complete graph, chain, or cycle, maximizing the sum of densities will generate the same bipartition as minimizing the ratio cut. If we restrict the cluster sizes to be all identical, then a maximum sum of densities solution will also be a minimum cut solution.

In the work which directly led to our approach, Kortsarz and Peleg [8] solved the MDSG problem for a graph using the "Provisioning Problem" formulation of Lawler [9].

The Provisioning Problem: Suppose there are n items $x_1, x_2, \ldots, x_n$ to select from, with selection of item x_i incurring cost $c_i > 0$. Suppose there are m sets of items $S_1, S_2, \ldots, S_m$ that are known to confer special benefits. A given item may be contained in several different sets. If all of the items in set S_j are selected, then a benefit $b_j > 0$ is obtained. The objective is to maximize { total benefit } $-$ { total cost } for the set of selected items.

Rhys [12] and Picard [10] showed that the Provisioning formulation can be transformed into a maximum flow problem in a network. The network is a bipartite graph with nodes n_{S_i} on one side representing the sets S_i, and nodes n_{x_j} on the other side representing the items x_j. A directed edge with infinite capacity is drawn from n_{S_i} to n_{x_j} if S_i contains x_j. The source is connected to each node n_{S_i} by an edge having capacity b_i, and the sink has a connection from each node n_{x_j} via an edge having capacity c_j. Edges in the minimum cut set which are incident from nodes in $\{n_{x_j}\}$ will correspond to the items selected in the optimum Provisioning solution.

We may extend the algorithm in [8] to solve the MDS problem for a hypergraph in polynomial time. The solution of the MDS problem via a series of Provisioning problem instances is as follows:

1. Let each node be an item, and let each hyperedge in the netlist be a set containing all items corresponding to its nodes.

2. Suppose we want to check whether there exists a subhypergraph with density d or higher, i.e., we seek a node subset U with $\frac{|E(U)|}{|U|} \geq d$ (here, $|E(U)|$ denotes the number of hyperedges completely contained in node set U). This may be restated as $|E(U)| - d|U| \geq 0$.

3. This is exactly the Provisioning formulation with the cost of each item being d and the benefit of each set being 1.

4. Since the difference between two distinct densities is at least $\frac{1}{n(n-1)}$ [6], we can use binary search to guess the maximum density. Only $O(\log n)$ flow computations are needed to determine the optimal solution.

This sequence of steps immediately yields our *Hyper-MDS* algorithm for the MDS formulation. Figure 3 shows the transformation from a hypergraph to a flow network instance. Note that the signal nets $N_1, \ldots, N_4$ correspond to the "sets".

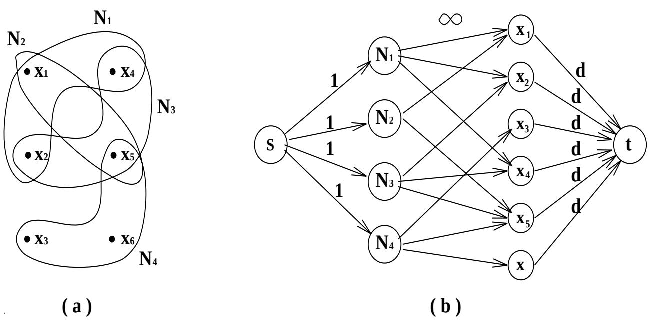

Figure 3: An example of a hypergraph (a) and its corresponding flow network instance (b). The hypergraph contains 6 nodes and 4 signal nets $N_1 = \{x_1, x_2, x_4\}$, $N_2 = \{x_1, x_5\}$, $N_3 = \{x_2, x_4, x_5\}$, and $N_4 = \{x_3, x_5, x_6\}$. We assign a "*guess*" density d to all edges incident from the nodes in the right side. When $d = \frac{3}{4}$, the edges $\{\overrightarrow{sN_4}, \overrightarrow{x_1t}, \overrightarrow{x_2t}, \overrightarrow{x_4t}, \overrightarrow{x_5t}\}$ will be in the minimum cut set, implying that the maximum density subhypergraph is over nodes $\{x_1, x_2, x_4, x_5\}$.

3 The k-Way Maximum Sum of Densities (k-MSD) Problem

In this section, we first show that the k-MSD problem is NP-hard. We then present a greedy heuristic for this problem which has performance ratio of 2 for the 2-MSD problem, and k for the k-MSD problem, but which in practice returns solutions that are remarkably close to optimal.

3.1 K-MSD is NP-Hard

Lemma 1 *The 2-MSD problem is NP-hard.*

Proof : The 2-MSD problem has equivalent complexity to minimizing the sum of 2 cluster densities (2-MinSumD). This is because the 2-MSD solution for a given graph G will be the same as the 2-MinSumD solution in the complement graph $\overline{G}$. To confirm that the 2-MSD problem is NP-hard, we show that Simple Max Cut[2] [5] reduces to 2-MinSumD.

Given graph $G(V, E)$, construct $G^*(V^*, E^*)$ with $2n$ nodes that contains G and a separate isomorphic copy $G'(V', E')$ of G. Every edge in $E \cup E'$ has unit capacity. Add one edge in G^* with capacity M between each corresponding pair ($v_i \in V$, $v_{i'} \in V'$). Note that for sufficiently large M, a 2-MinSumD partitioning $(X, \bar{X})$ in G^* must have $|X| = |\bar{X}|$ with either $v_i \in X$ and $v_{i'} \in \bar{X}$, or $v_i \in \bar{X}$ and $v_{i'} \in X$, for all $1 \le i \le n$: only such a partitioning can have 2-MinSumD value as small as $\frac{2|E|}{n}$. A partitioning $(X, \bar{X})$ that solves 2-MinSumD in G^* will have sum of densities $= \frac{2|E|-2k}{n}$, where k is the value of a maximum cut of G, and $(X \cap V, \bar{X} \cap V)$ will be a maximum cut in the original graph G. (See Figure 4.) □

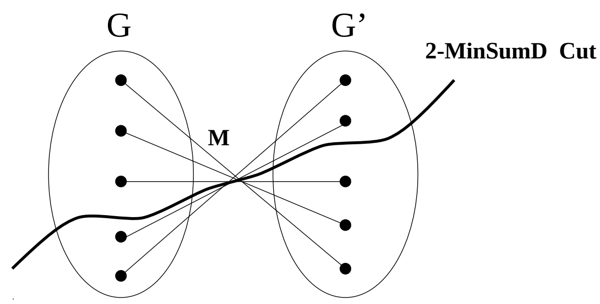

Figure 4: The construction of G^*

Theorem 1 *The k-MSD problem is NP-hard for all fixed $k \ge 3$.*

Proof : For any $k \ge 3$ and any given graph G, we construct a graph G^* containing G and $k-2$ clusters which each have very high density D. For sufficiently large D, the k-MSD solution for G^* must contain the $k-2$ dense clusters and two clusters from G. We thus obtain a reduction from the 2-MSD problem. □

3.2 A Good k-MSD Heuristic

A simple heuristic algorithm for the k-MSD problem is as follows.

- First, we obtain a decomposition into up to k clusters via a scheme called *MDS_Peeling*(Figure 5), which iteratively applies the *Hyper-MDS* algorithm. The idea is to peel off a maximum-density cluster $k-1$ times, or until all nodes in the graph have been exhausted.

- Second, we compute a linear ordering of the vertices of G, based on the $\le k$ clusters obtained from MDS_Peeling. Essentially, we place these k clusters in the order that they were found, and apply the WINDOW ordering of [2] within each cluster to place each node into the overall ordering.

- Third, we apply dynamic programming to efficiently split the linear ordering into a k-way partitioning that is optimal, subject to the constraint that each cluster is contiguous in the ordering [1].

We thus obtain our **MDS_Peeling+DPRP** algorithm. As we shall see, this method implicitly gives an upper bound for the k-MSD solution, and has good performance in practice.

Observation 1 *Given a hypergraph $H(V, E)$, let D be the density of the maximum density subgraph of H found by the MDS_Peeling phase. Then, an upper bound on the sum of densities S in the optimal k-MSD solution is $S \le kD$, and this bound is tight.*

[2] Simple Max Cut is stated as follows. Instance: Graph $G = (V, E)$, positive integer K. Question: Is there a partition of V into disjoint sets V_1 and V_2 such that the number of edges in E having one endpoint in V_1 and one endpoint in V_2 is at least K?

MDS_Peeling Algorithm
Input : Hypergraph $H(V, E)$, and k
Output: k clusters $cluster_1, cluster_2, \ldots, cluster_k$

```
i = 0;
while (E is not empty) and (i < k) do
    Find the maximum density subhypergraph
        U(V', E') of H
    Let E'' be the hyperedge cut set of (V', V - V')
    E = E - E' - E'';
    cluster_i = V'
    V = V - V'
    i = i + 1;
if E is not empty then
    cluster_k = V
```

Figure 5: MDS_Peeling algorithm to generate clusters.

Proof : The maximum-density cluster of H has density D, so the sum of densities of k clusters can be at most kD. If H consists of k disjoint clusters each with density D, this upper bound is tight. $\square$

Observation 2 *The MDS_Peeling+DPRP has performance ratio* 2 *for the* 2*-MSD problem.*

Proof : Let D be the density of the maximum-density cluster of H, and let D' be the density of the remaining cluster. The sum of densities returned by MDS_Peeling+DPRP is $\geq \frac{D+D'}{2D} = \frac{1}{2} + \frac{D'}{2D} \geq \frac{1}{2}$. $\square$

In general, MDS_Peeling+DPRP has a tight performance ratio of k for the k-MSD problem. However, its performance seems much closer to optimal in practice.

3.3 Results for MDS_Peeling + DPRP

We have found heuristic k-MSD solutions using both MDS_Peeling+DPRP as well as the obvious variant of the Fiduccia-Mattheyses (FM) method. Our test cases consist of standard partitioning benchmarks maintained by ACM SIGDA. Our FM variant, which we call *Density-FM*, is standard k-way FM with gains updated based on the sum of densities metric; our implementation adapts code from [13]. Table 1 compares MDS_Peeling+DPRP against the best result of 10 Density-FM runs. MDS_Peeling+DPRP averages 5.29% improvement over Density-FM.

We also note that MDS_Peeling+DPRP can perform quite well when judged against the theoretical upper bound on the sum of densities. Table 2 gives a more careful study of MDS_Peeling+DPRP results for k-way clustering of the Primary1 and Primary2 benchmarks with a range of k values; we compare the sum of densities with the theoretical upper bound for the k-MSD solution. (Recall that the upper bound is k times the largest subgraph density.) Although MDS_Peeling has a worst-case performance ratio of k for k-MSD, the results show that it can actually be very close to optimum in practice. (For three test cases with very small, very dense subhypergraphs – Test02, Test04, Test05 – this analysis fails since kD is a very loose upper bound.)

4 Density-Based Ratio Cuts

We close by noting a variation of Hyper-MDS which yields good 2-way ratio cut solutions. Such a result is interesting because it shows that "density can capture cut", while the converse would seem difficult.

Recall that the Hyper-MDS algorithm will return the densest subhypergraph of the netlist hypergraph, where each signal net has "credit" = 1, and each node has "cost" = 1. This has the following weaknesses with respect to minimizing the ratio cut:

- the method concentrates on collecting many smaller nets into the cluster, and ignores resulting growth of the cluster boundary (i.e., nets cut).
- the method does not distinguish between credit for nets with high degree and credit for nets with low degree; however, a high-degree net will potentially cause more cuts.
- the method does not distinguish between cost of nodes with high degree and cost of nodes with low degree; however, a high-degree node will potentially cause more cuts.

To create a bias against nets and nodes with large degree, we can adjust the *net credit* for each net and the *node cost* for each node. Details of the resulting heuristic, and experimental results showing that the density-based approach is very competitive with the best known ratio cut methods, are given in [7].

Ckts	#Clusters	Upper Bound	MDS_Peeling + DPRP	% of optimum
Prim1	8	10.00	9.32	93.20%
	10	12.50	11.27	90.16%
	15	18.75	16.09	85.81%
	20	25.00	20.49	81.96%
Prim2	8	8.06	7.80	96.77%
	10	10.08	9.65	95.73%
	15	15.11	14.65	96.96%
	20	20.15	19.31	95.83%
	25	25.18	23.96	95.15%
	30	30.22	28.55	94.47%
	35	35.26	33.01	93.62%
	40	40.30	37.38	92.75%

Table 2: Comparison of sum of cluster densities in the MDS_Peeling+DPRP solution versus the theoretical k-MSD upper bound, for large values of k. The 20-way clustering for Primary1 (833 nodes) requires 110 seconds and the 40-way clustering for Primary2 (3014 nodes) requires 968 seconds on a Sun SPARC-10. Note that we only need to run MDS_Peeling+DPRP once to obtain results for all values of k.

5 Conclusions

We have introduced a new sum of densities objective that can lead to more natural circuit decompositions than previous objectives. Our objective can furthermore be used to unify top-down cut-based partitioning with bottom-up density-based clustering. We have given an optimal Provisioning-based flow solution, called Hyper-MDS, for the MDS problem; we have also proposed the MDS_Peeling + DPRP heuristic for the k-MSD problem. Related density-based formulations which may be of future interest include the following.

Test Cases	Algorithm	Sum of Densities							Average Improvement
		$k=2$	$k=3$	$k=4$	$k=8$	$k=10$	$k=15$	$k=20$	
Prim1	MDS_Peeling+DPRP	2.31	3.50	4.69	9.32	11.27	16.09	20.49	4.98%
	Density-FM	2.31	3.33	4.51	8.68	10.16	15.36	19.95	
Prim2	MDS_Peeling+DPRP	1.88	2.80	3.80	7.80	9.65	14.65	19.31	2.15%
	Density-FM	1.96	2.87	3.95	7.25	9.25	13.57	18.30	
Test02	MDS_Peeling+DPRP	15.49	29.95	31.15	35.55	37.60	42.53	47.23	0.93%
	Density-FM	15.49	29.95	31.07	35.17	37.24	41.74	46.17	
Test03	MDS_Peeling+DPRP	4.47	5.97	7.46	12.59	14.83	19.85	24.76	23.51%
	Density-FM	2.54	3.69	7.11	12.07	13.57	19.06	23.78	
Test04	MDS_Peeling+DPRP	15.55	27.01	28.33	33.47	35.82	41.47	46.73	1.62%
	Density-FM	15.55	27.01	28.31	32.56	34.72	40.07	45.91	
Test05	MDS_Peeling+DPRP	16.53	30.01	31.45	36.99	39.57	45.50	51.06	3.66%
	Density-FM	15.51	30.01	31.30	35.72	37.58	42.46	48.98	
Test06	MDS_Peeling+DPRP	1.89	2.85	3.77	7.57	9.41	13.82	18.27	0.22%
	Density-FM	1.90	2.90	3.82	7.56	9.19	13.52	18.20	
avq.small	MDS_Peeling+DPRP	2.17	3.22	4.18	7.97	9.91	14.61	19.19	—
	Density-FM	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	

Table 1: Comparison of sum of cluster densities for MDS_Peeling+DPRP and for the best of 10 Density-FM runs. Results for avq.small benchmark using Density-FM would require many days of SPARC-10 time.

- **Bounded Size Maximum Density Subhypergraph (BMDS)** problem : Given a hypergraph $H(V,E)$ and an integer B, find the subhypergraph of H with maximum density and size $\leq B$.

  While MDS was polynomial time solvable, BMDS is shown NP-complete by reduction from Maximum Clique.

- **Max-Density Subhypergraph with Prescribed Node** problem : Given a hypergraph $H(V,E)$ and a prescribed node p, find the maximum density subhypergraph which contains node p.

  This can be solved using the $0-1$ fractional programming technique in [11] by assigning the variable corresponding to p to 1, and transforming the resulting fractional expression to a series of flow computations.

- **Max-Density Subhypergraph with Excluded Node** problem : Given a hypergraph $H(V,E)$ and a prescribed node p, find the maximum density subhypergraph which does not contain node p.

  This problem can be solved in the same time complexity as the MDS problem. We can remove node p and all edges incident from p, and solve the MDS problem in the remaining hypergraph.

6 Acknowledgments

We are grateful to Dr. Lars Hagen for his valuable comments, and to Professor Laura Sanchis for providing k-way FM code [13]. Also, thanks to Dr. L. T. Liu for valuable discussion.

References

[1] C. J. Alpert and A. B. Kahng, "Multi-Way Partitioning Via Spacefilling Curves and Dynamic Programming", 31th *ACM/IEEE Design Automation Conference*, 1994, pp. 652-657.

[2] C. J. Alpert and A. B. Kahng, "A General Framework for Vertex Orderings, With Applications to Netlist Clustering", *Proc. IEEE Intl. Conf. on Computer-Aided Design*, 1994, to appear.

[3] J. Cong and M. Smith, "A Parallel Bottom-up Clustering Algorithm with Applications to Circuit Partitioning in VLSI Design", 30th *ACM/IEEE Design Automation Conference*, 1993, pp. 755-760.

[4] C. M. Fiduccia and R. M. Mattheyses, "A Linear Time Heuristic for Improving Network Partitioning", *19th ACM/IEEE Design Automation Conference*, 1982, pp. 175-181.

[5] M. R. Garey and D. S. Johnson, *Computers and Intractability : A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, 1979.

[6] A. V. Goldberg, "Finding a Maximum Density Subgraph", *Technical Report No. UCB/CSD 84/171*, May 1984.

[7] Dennis J.-H. Huang and Andrew B. Kahng, "When Clusters Meet Partitions: New Density-Based Methods for Circuit Decomposition", UCLA Computer Science Department, TR-940019, 1994.

[8] G. Kortsarz and D. Peleg, "Generating Sparse 2-spanners", *Third Scandinavian Workshop Proceedings*, 1992, pp. 73-82.

[9] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids* Holt, Rinehart and Wiston, New York, 1976.

[10] J.-C. Picard, "Maximal Closure of a Graph and Application to Combinatorial Problems", *Management Science*, Vol. 22, 1976, pp. 1268-1272.

[11] J.-C. Picard and M. Queyranne, "A Network Flow Solution to Some Nonlinear 0-1 Programming Programs, with Applications to Graph Theory", *Networks*, Vol. 12, 1982, pp. 141-159.

[12] J. M. W. Rhys, "A Selection Problem of Shared Fixed Costs and Network Flows", *Management Science*, Vol. 17, 1970, pp. 200-207.

[13] L. A. Sanchis, "Multiple-Way Network Partitioning", *IEEE Trans. Computers*, vol. 38, no. 1, 1989, pp. 62-81.

[14] Y. C. Wei and C. K. Cheng, "Ratio Cut Partitioning for Hierarchical Designs", *IEEE Trans. Computer-Aided Design*, vol. 10, July 1992, pp. 911-921.