Improving Graph Partitions using Submodular Functions

Sachin B. Patkar [1] H.Narayanan [2]

Abstract

We investigate into the role of submodular functions in designing new heuristics and approximate algorithms to some NP-hard problems arising in the field of VLSI Design Automation. In particular we design and implement efficient heuristic for improving a bipartition of a graph in the sense of ratioCut ([2,14]). We also design an approximate algorithm for another NP-hard problem which is a dual of the well-known NP-hard problem of finding a densest k-subgraph of a graph (see [1,7]). Our algorithms are based on submodular function and are implementable in polynomial time using efficient network flow based subroutines. To the best of our knowledge our algorithms are the first ones to use submodular functions based approach for the problems considered here. We also describe the experimental results which provide the evidence of our heuristic for improving the ratioCut.

1 Introduction

Partitioning the vertex set of a graph is of immense importance in many applications such as circuit partitioning. Circuit partitioning is the task of dividing a circuit into smaller components. Such requirements arise naturally in various VLSI applications, e.g. to divide a circuit into blocks that are then implemented on separate components or modules, such as printed circuit boards, FPGAs or chips [15]. The objective is to divide the circuit into blocks such that each component falls within prescribed sizes and the complexity of connections between these components is reduced. Partitioning is also of critical importance in many other fields. In this paper, we mainly present new theoretical and experimental results of a strategy that aims at improving a given partition of the vertex set into 2 parts.

[1] Department of Mathematics, Indian Institute of Technology, Bombay, Mumbai-400 076, India. e-mail: patkar@math.iitb.ac.in
[2] Department of Electrical Engineering, Indian Institute of Technology, Bombay, Mumbai-400 076, India. e-mail: hn@ee.iitb.ac.in

Preprint submitted to Elsevier Science *23 March 2002*

Furthermore there are several natural and practical situations, in which given a "good" partition in the sense of "small" interconnections, we may still be required to meet the constraints that each block should not have more than certain number of interconnections. One may "prune" the blocks by removing appropriate parts of each block of the partition such that the interconnection constraints are met. The removed parts of the blocks can likely be economically accommodated on cheaper modules of programmable logic devices. A natural question arises: how to find a subset of vertices of a block so that the required improvement in the interconnections results on removal of the above subset from the block? In this paper we also discuss the related problem of identifying a subset of k vertices (k specified a priori!) of a graph such that the size of the set of edges incident on these vertices is minimum. Such a subset would be a good candidate for being removed from the block. We will show that this problem is *NP-hard* and proceed to give a 2-approximate polynomial time algorithm for it. We can also recognize if the solution output by this algorithm for an instance is indeed an optimal one, too!

We use the following notation. $G(V, E, w_v, w_e)$ denotes an undirected graph with vertex set (or node set) V, edge set E and *positive* weight assignments $w_v(.)$ and $w_e(.)$ on the vertex set and edge set respectively.

Remark 1 *Throughout this paper, the node weights $w_v(.)$ and the edge weights $w_e(.)$ are strictly positive unless otherwise mentioned.*

We use $\delta(U)$ to denote the set of edges having exactly one endpoint in the vertex subset $U \subseteq V$. The cut cost of a block V' is $w_e(\delta(V'))$. The cut cost of a partition Π of V is defined as $\frac{1}{2} * \sum_{N\in\Pi} w_e(\delta(N))$. Multi-way graph partitioning (i.e. computing a partition into k blocks whose cut-cost is smallest) is known to be NP-hard [12].

We call (V_1, V_2) a bipartition of V if $V_1, V_2 \neq \emptyset$, $V_1 \cap V_2 = \emptyset$ and $V_1 \cup V_2 = V$. 2-way partitioning (also called "bipartitioning") of graphs is a well researched problem as many of the practical k-way partitioning strategies make use of bipartitioning in a recursive manner.

Although typically the bipartitions are required to be balanced (in the sense of both the parts having similar weights of their vertex sets), very often it may be desired to have a possibly unbalanced partition that is more "natural" (in the sense of the small value of ratio of the cut value of a block to its weight). The above mentioned "naturalness" is captured by the well-known concept of *ratioCut* (see [2,14]).

The ratioCut for $\emptyset \neq W \subset V$ is defined as follows:

$$\text{ratioCut}(W) = \frac{w_e(\delta(W))}{w_v(W) * w_v(V - W)}.$$

Note that the notion of the *ratioCut* as defined above is really associated with the corresponding bipartition. We also call $ratioCut(W)$ as the *ratioCut* of the corresponding bipartitions, i.e. $(W, V - W)$ or $(V - W, W)$. In [2], the authors use the concept of "scaled cost" which specializes to the concept of "ratioCut" for 2-way partition.

The problem of finding a bipartition into equal sized vertex subsets that minimizes the cut-cost is *NP-hard* (see for details [15]). Similarly the problem of finding a bipartition, say (V_1, V_2) of the node set V, that minimizes the $ratioCut(V_1)$ is also *NP-hard* (see for details [14]).

We present a new global approach for the problem of improving upon a given bipartition in the sense of smaller ratioCut. Motivation and justification is based on submodular function minimization and the Principal Partition approach (cf. [8,16]). In particular, we formulate and solve a ratio-optimization problem and solve it using theory of Principal Partition. We may note that ratio-optimization problems have been widely researched in several different contexts (see [4,8,10,13,18]).

Earlier work in this area typically used graph theoretic approach based on iterative improvements (see for details [15]). A survey of heuristics based on graph theoretic or combinatorial approach may be found in [15]. Spectral methods are also popular (see for instance, [2,3]). The problem of r-way partitioning has already been theoretically investigated in [17] using submodular function based approaches.

Fiduccia-Matheyses (F-M) method (see for details [15]) and its variants are among the most popular and effective methods for the problem of improving bipartitions of a graph. They attempt improvement in each iteration by doing a local search. They typically make use of the following kind of information: $Gain(u) =$ the amount of reduction in the cut value if u is moved across to the other block. Thus we note that the logic that is based on the above concept has its limitations and methods such F-M method tend to stall at a local optimum.

We investigate a combinatorial generalization of the concept of $Gain(.)$ to arbitrary subsets of V_1 or V_2. Given a graph $G(V, E, w_v, w_e)$, and a bipartition (V_1, V_2), we define for a subset U of either V_1 or V_2, $GGain(U) =$ the amount of reduction in the cut value if U is moved across to the other block.

We shall show in this paper that the concept of $GGain$ (read, "generalized-gain") leads us to a formulation involving minimization over submodular functions for which efficient algorithms are known. We mainly use ideas from the theory of Principal Partition. The investigations of this field are described in [8,16]. We also use ideas from the literature on "fractional programming" [5].

Identifying Weakly linked Subset of Vertices

Now we formally describe the problem that has implications in devising strategies for pruning blocks of a given partition. We begin with some more notation. $E(U)$ is defined as the set of edges having both the endpoints in U. $\Gamma(U)$ is defined as the set of edges having at least one of the endpoints in U. Note that $\Gamma(U) = E - E(V - U)$. The problem of finding **densest k-subgraph** is to find, for a given k, a subset $U \subseteq V$, with $|U| = k$, that maximizes $|E(U)|$. This problem is **NP-hard** as **clique-decision problem** can be reduced to it. Indeed, this problem of finding a densest k-subgraph remains NP-hard under several simplifying assumptions about the graph. Approximate algorithms are not yet known for this problem in the case of general graphs (see [1,7]).

The problem of finding a **weakest linked k-vertex subset** is to find, for a given k, a subset $U \subseteq V$, with $|U| = k$, that minimizes $|\Gamma(U)|$. Noting the duality between $\Gamma(.)$ and $E(.)$ it is clear that this problem is **NP-hard**, too.

We shall demonstrate the use of submodular functions in designing an approximate algorithm for the **weakest linked k-vertex subset** problem. This is significant in view of the above mentioned absence of approximate algorithms for dual problem of finding a densest k-subgraph.

2 Preliminaries and Notation

Let $f : 2^V \longrightarrow \mathbf{R}$ be a given function. f is said to be *normalized* if $f(\emptyset) = 0$. The function f is submodular **iff**
$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y) \ldots \forall\, X, Y \subseteq V$. f is supermodular **iff** $-f$ is submodular. And f is modular **iff** it is both submodular and supermodular.

The function $|\delta(.)| : 2^V \longrightarrow \mathbf{R}$ is a submodular function. The function $w_e(\delta(.)) : 2^V \longrightarrow \mathbf{R}$ is also a submodular function as $w_e \geq 0$. Let $f_\lambda(.)$ denote the function $w_e(\delta(.)) - \lambda * w_v(.)$. The function $|\Gamma(.)|$ is also submodular and the function $|E(.)|$ is supermodular.

Let $g : 2^S \longrightarrow \mathbf{R}$ be a submodular (supermodular) function and let $w(\cdot)$ be a weight function on S. Then the Principal Partition of (g, w) (or of g, if w is clear from the context), is the collection of all sets which minimize (maximize) $g(X) - \lambda * w(X), X \subseteq S$, λ real.

For more details on submodular functions and the theory of Principal Partition, the reader is referred to [6,8,16].

3 A Formulation Involving Optimization of Submodular Functions

We begin with a following straightforward modeling of the concept of $GGain(.)$ using the well-known submodular function $w_e(\delta(.))$.

Proposition 2 *Let (V_1, V_2) be a bipartition of $G(V, E, w_v, w_e)$. For $U \subseteq V_1$, $GGain(U) = w_e(\delta(V_1)) - w_e(\delta(V_1 - U))$. Similarly for $U \subseteq V_2$, $GGain(U) = w_e(\delta(V_2)) - w_e(\delta(V_2 - U))$.*

For a nonempty U that is a subset of either V_1 or V_2, we define **averageGain** in a natural way as follows: $averageGain(U) = GGain(U)/w_v(U)$.

Problem 3 *We consider the natural problem of finding nonempty $\hat{U} \subseteq V_1$ that maximizes the* averageGain *over all nonempty subsets of V_1. Similarly, we may consider the same problem on V_2.*

Definition 4 *Given a bipartition (V_1, V_2), let $\mu(G(V, E, w_v, w_e), V_1)$ denote the maximum averageGain attained over the nonempty subsets of V_1. Similarly we define $\mu(G(V, E, w_v, w_e), V_2)$. We use μ as an abbreviation whenever the graph and the block of the bipartition is clear from the context.*

Since the above problem is a ratio-optimization problem, we borrow the relevant ideas from [5] and [8] and consider the following problem:

Problem 5 *For each real λ, find subsets that solve*

$$\min_{U \subseteq V_1} (w_e(\delta(U)) - \lambda * w_v(U)).$$

The above problem is that of computing the **Principal Partition of** $(w_e(\delta(.)), w_v(.))$ Similar problem may be defined for V_2. We shall focus w.l.o.g. on V_1 only. The case of V_2 is symmetrical.

Proposition 6 $\hat{\lambda} = \mu(G(V, E, w_v, w_e), V_1)$ **iff** *$\hat{\lambda}$ is such that $\exists$ a proper subset $Z \subset V_1$ satisfying*

$$\begin{aligned}
&\min_{U \subseteq V_1} \quad (w_e(\delta(U)) - \hat{\lambda} * w_v(U)) \\
&\qquad = w_e(\delta(V_1)) - \hat{\lambda} * w_v(V_1) \\
&\qquad = w_e(\delta(Z)) - \hat{\lambda} * w_v(Z).
\end{aligned}$$

(Proof of Proposition 6:) (*Only If part:*) Let $\emptyset \neq W \subseteq V_1$ be such that

$$\hat{\lambda} = \mu = \frac{w_e(\delta(V_1)) - w_e(\delta(V_1 - W))}{w_v(W)}.$$

We claim that $Z = V_1 - W$ proves the "only if" part of this proposition. Indeed,

$$w_e(\delta(V_1)) - \hat{\lambda} * w_v(V_1) = w_e(\delta(V_1 - W)) - \hat{\lambda} * w_v(V_1 - W) \tag{1}$$

Furthermore, for any $U \subset V_1$,

$$\hat{\lambda} = \frac{w_e(\delta(V_1)) - w_e(\delta(V_1 - W))}{w_v(W)} \geq \frac{w_e(\delta(V_1)) - w_e(\delta(U))}{w_v(V_1 - U)},$$

thus implying

$$(w_e(\delta(V_1)) - \hat{\lambda} * w_v(V_1)) \leq (w_e(\delta(U)) - \hat{\lambda} * w_v(U)). \tag{2}$$

Thus it follows from (1) and (2) that the claim $Z = V_1 - W$ proves "only if" part of this proposition.

(*If part:*) Let $\hat{\lambda}$ and $Z \subset V_1$ be as in the hypothesis. Need to show that $\hat{\lambda} = \mu$. Note that for any $\emptyset \neq W$,

$$(w_e(\delta(V_1)) - \hat{\lambda} * w_v(V_1)) \leq (w_e(\delta(V_1 - W)) - \hat{\lambda} * w_v(V_1 - W)) \tag{3}$$

implying

$$\frac{w_e(\delta(V_1)) - w_e(\delta(V_1 - W))}{w_v(W)} \leq \hat{\lambda} = \frac{w_e(\delta(V_1)) - w_e(\delta(Z))}{w_v(V_1 - Z)}.$$

Thus $\hat{\lambda}$ is indeed equal to $\mu(G(V, E, w_v, w_e), V_1)$. $\square$

Now we shall note the significance of the concept of maximum averageGain for designing a strategy to improve the ratioCut.

Proposition 7 *Let $G(V, E, w_v, w_e)$ be a graph with a given bipartition $(V_1, V - V_1)$. Let $\hat{U}$ be a nonempty proper subset of V_1 such that $\mu(G(V, E, w_v, w_e), V_1) = \frac{w_e(\delta(V_1)) - w_e(\delta(V_1 - \hat{U}))}{w_v(\hat{U})}$. Then $ratioCut(V_1 - \hat{U}) < ratioCut(V_1)$.*

(Proof of Proposition 7:)

$$\frac{w_e(\delta(V_1)) - w_e(\delta(V_1 - \hat{U}))}{w_v(\hat{U})} \geq \frac{w_e(\delta(V_1)) - w_e(\delta(V_1 - V_1))}{w_v(V_1)}.$$

This may be rewritten as

$$\frac{w_e(\delta(V_1)) - w_e(\delta(V_1 - \hat{U}))}{w_v(V_1) - w_v(V_1 - \hat{U})} \geq \frac{w_e(\delta(V_1))}{w_v(V_1)}.$$

The above implies $\frac{w_e(\delta(V_1 - \hat{U}))}{w_v(V_1 - \hat{U})} \leq \frac{w_e(\delta(V_1))}{w_v(V_1)}$. This clearly implies that $ratioCut(V_1 - \hat{U}) < ratioCut(V_1)$, as $w_v(V - V_1) < w_v(V - (V_1 - \hat{U}))$. $\square$

The proposition may also be seen to follow as a corollary to Lemma 11. Thus the submodular function optimization introduced above helps in improvement in the ratioCut in a "global manner" using the generalized concept of $GGain()$, as compared to the improvements attempted by strategies which make use of the "local" concept of $Gain(.)$.

In the later sections we shall see that the above formulation indeed gives a lot more information useful for improving the ratioCut.

4 An Algorithm to Improve RatioCut

In this section, we present and justify a new algorithm for improving a 2-way circuit partitioning in the sense of the well-known measure of *ratioCut*. We will need to know the following relevant properties [8,16] of Principal Partition to justify our scheme.

4.1 Some Relevant Properties of Principal Partition

Recall the definition of Principal Partition given in section 2. We shall consider, in particular, the Principal Partition of $(w_e(\delta(.)), w_v(.))$ (with the function $w_e(\delta(.))$ defined on the subsets of V_1). Recall that f_λ denotes the function $w_e(\delta(.)) - \lambda * w_v(.)$, defined on subsets of V_1.

Some properties of the Principal Partition which are relevant to this paper are as follows: [8,16]:

(1) There is a unique maximal set X^λ and a unique minimal set X_λ at which f_λ reaches the minimum over the subsets of V_1. We call these sets *critical sets* in the Principal Partition of $(w_e(\delta(.)), w_v)$ for $G(V, E, w_v, w_e)$ and $V_1 \subseteq V$. Thus every subset that minimizes f_λ is contained in X^λ and contains X_λ.
(2) If $\lambda_1 < \lambda_2$, it can be shown that $X^{\lambda_1} \subseteq X_{\lambda_2}$. Thus all the critical sets form a "nested" sequence w.r.t. inclusion.
(3) For not more than $|V_1|$ values of λ, $X_\lambda \neq X^\lambda$. Such values are called *critical values* in the above Principal Partition of $(w_e(\delta(.)), w_v)$.
(4) Let $\lambda_1 < \lambda_2 < \lambda_3 \ldots < \lambda_t$ be the sequence of all critical values.
Then $X^{\lambda_i} = X_{\lambda_{i+1}}$ for $i = 1, 2, \ldots, t-1$, and $X_{\lambda_1} = \emptyset$, $X^{\lambda_t} = V_1$. The sequence $X_{\lambda_1} \subset X_{\lambda_2} \subset \ldots \subset X_{\lambda_t} \subset X^{\lambda_t}$ is called the **Principal Sequence** of function $w_e(\delta(.))$ for $G(V, E, w_v, w_e)$ and $V_1 \subseteq V$.

Noting the definition of the largest critical value in the above mentioned Principal Sequence, we get the following corollary to Proposition 6. We also note that the following result may be considered implicit in the general result on ratio-optimization by Fujishige [8].

Corollary 8 *$\mu(G(V, E, w_v, w_e), V_1)$ is equal to the largest critical value in the Principal Partition of $(w_e(\delta(.)), w_v(.))$ for $G(V, E, w_v, w_e)$ and $V_1 \subseteq V$.*

We may also note the following trivial observation:

Proposition 9 *All the critical values associated with the Principal Partition of $(w_e(\delta(.)), w_v(.))$ for $G(V, E, w_v, w_e)$ and $V_1 \subseteq V$ are non-negative.*

(Proof of Proposition 9:)
Note that due to the positivity of the weighs, $\emptyset$ solves $\min_{U \subseteq V_1}(w_e(\delta(U)) - \lambda * w_v(U))$ for $\lambda = 0$. One of the properties of the Principal Partition being considered here states that $\lambda_1 < \lambda_2$ implies $X^{\lambda_1} \subseteq X_{\lambda_2}$. This clearly implies that for $\lambda < 0$, $\min_{U \subseteq V_1}(w_e(\delta(U)) - \lambda * w_v(U))$ is attained only at $\emptyset$. Thus a negative value cannot be a critical value. $\square$

4.2 The Algorithm

A high level description of our algorithm that attempts to improve ratioCut is as follows:

(1) Let V_1, V_2 denote the two blocks of the given bipartition.
- Compute the Principal Sequence of $w_v(\delta(.))$ for the given $G(V, E, w_v, w_e)$ and $V_1 \subseteq V$.

- Compute the Principal Sequence of $w_v(\delta(.))$ for the given $G(V, E, w_v, w_e)$ and $V_2 \subseteq V$.

(2) For each of the nonempty subsets, say X, in either of the above two Principal Sequences, if $(X, V-X)$ is not the same as (V_1, V_2) and (V_2, V_1) then **output** $(X, V-X)$ as a bipartition whose ratioCut is strictly lower (i.e. better) than that of (V_1, V_2).

(3) For each pair of nonempty subsets (one from each of the above two Principal Sequences), compute a minimum cut separating these two subsets. Let $(Y, V-Y)$ be the minimum cut obtained. **Output** $(Y, V-Y)$ if it is not already output and is different from (V_1, V_2) and (V_2, V_1). The ratioCut of the above bipartition $(Y, V-Y)$ is strictly lower than that of (V_1, V_2).

We note that the above algorithm may fail to find an improved bipartition, as there exists a possibility of non-existence of any nonempty proper subsets (of V_1 and V_2, respectively) in either of the two Principal Sequences used in the above algorithm. However, later in this paper, we show that in most of our experiments with benchmark graphs, this algorithm did manage to find at least one improved bipartition.

4.3 Computing the Required Principal Sequence

Specializing the generic algorithm to compute the Principal Sequence of a given submodular function [16], we can compute the Principal Sequence of $w_e(\delta(.))$ for $G(V, E, w_v, w_e)$ and $V_1 \subseteq V$, by $O(|V_1|)$ invocations of the subroutine that computes the smallest or largest subset that minimizes $f_\lambda(.)$ over the subsets of V_1.

We shall now define a flow network that will be used to find the smallest set that minimizes $f_\lambda(.)$ over the subsets of V_1. We also recall that all the critical values in this Principal Partition are non-negative (see Proposition 9).

Our flow network model is as follows: Given a graph $G(V, E, w_v, w_e)$, a subset $V_1 \subseteq V$ and a real non-negative number λ, we define a flow network $FL(V, E, w_v, w_e, V_1, \lambda)$ as follows:

$FL(V, E, w_v, w_e, V_1, \lambda)$ consists of nodes representing source s, sink t (with $s, t \notin V$) and nodes in V_1. The arcs of the above network are as follows:
$\{(s, u), (u, t) \mid u \in V_1\}$ and
$\{(u, v), (v, u) | u, v \in V_1$ are connected by an edge in $G\}$.

The capacities on arcs are as follows,
$cap(s, u) = \lambda * w_v(u)\ \forall u \in V_1$
$cap(u, t) = \sum_{(u,v) \in E, u \in V_1, v \notin V_1} w_e((u, v))\ \forall u \in V_1$

$cap(u, v) = w_e(e')$ where e' connects u, v in G.

Any cut partitions the node set of the flow network in two parts, one of which has the source as one of its members and the other has the sink as one of its members. The set of nodes containing the source (sink) is called s-part (t-part) of the cut.

Given $U \subseteq V_1$ we define in $FL(V, E, w_v, w_e, V_1, \lambda)$, $cut(U)$ as the cut whose s-part is $\{s\} \cup U$.

To find the smallest set that minimizes $f_\lambda(.)$ over the subsets of V_1, a maximum flow (see [11]) in the network $FL(V, E, w_v, w_e, V_1, \lambda)$ is constructed. Then the minimum cut that has the smallest s-part is constructed. For this purpose, a breadth first search that starts at the source s and which traverses all the unsaturated arcs in the forward direction and all arcs with nonzero flow in the reverse direction, is performed. Similar approach would yield the minimum cut with the largest s-part (general results on the structure of minimum cuts separating a source and a sink are due to Picard and Queyeranne, see [16,19] for details).

The following simple lemma justifies our approach.

Lemma 10 *Let W be a subset of V_1. Then $capacity(cut(W)) = f_\lambda(W) + \lambda * w_v(V_1)$.*

Thus we have a subroutine that computes the smallest or largest subset that minimizes $f_\lambda(.)$ over subsets of V_1. We can therefore compute the Principal Sequence of $w_e(\delta(.))$ for $G(V, E, w_v, w_e)$ and $V_1 \subseteq V$, by $O(|V_1|)$ invocations of the above subroutine on flow-networks with $O(|V_1|)$ nodes and $O(|E|)$ arcs.

But indeed, a straightforward application of the well-known results of [9] on parametric network flow computations one can see that the Principal Sequence of $w_e(\delta(.))$ for $G(V, E, w_v, w_e)$ and $V_1 \subseteq V$ can be computed in time $O(|V_1||E| \log(|V_1|^2/|E|))$, i.e. in time required for a single maximum flow computation.

4.4 Justification of Algorithm

We first establish that any critical set in the Principal Sequence that is nonempty and a proper subset of V_1 will "help us" in improving the ratio-Cut. Indeed, the following lemma says that every nonempty proper subset of V_1 in the Principal Partition, being considered here, will help in improving the ratioCut.

Lemma 11 *Let $G(V, E, w_v, w_e)$ be a given graph and let $V_1 \subseteq V$. Let U' be a subset that minimizes $w_e(\delta(.)) - \lambda * w_v(.)$ over the subsets of V_1. If $\emptyset \neq U' \subset V_1$ then,*

$$\text{ratioCut}(U') < \text{ratioCut}(V_1).$$

(Proof of Lemma 11 :)
We have

$$0 = w_e(\delta(\emptyset)) - \lambda * w_v(\emptyset) \geq w_e(\delta(U')) - \lambda * w_v(U') \leq w_e(\delta(V_1)) - \lambda * w_v(V_1) \tag{4}$$

From the above we get,

$$(\lambda - \frac{w_e(\delta(U'))}{w_v(U')}) \geq 0. \tag{5}$$

Rewriting part of equation 4 we get,

$$(\lambda - \frac{w_e(\delta(U'))}{w_v(U')}) * w_v(U') \geq (\lambda - \frac{w_e(\delta(V_1))}{w_v(V_1)}) * w_v(V_1). \tag{6}$$

As $(\lambda - \frac{w_e(\delta(U'))}{w_v(U')}) \geq 0$ and $w_v(V_1) > w_v(U') \geq 0$ (due to positivity of weights), we get

$$(\lambda - \frac{w_e(\delta(U'))}{w_v(U')}) \geq (\lambda - \frac{w_e(\delta(V_1))}{w_v(V_1)}) \tag{7}$$

thus implying,

$$\frac{w_e(\delta(U'))}{w_v(U')} \leq \frac{w_e(\delta(V_1))}{w_v(V_1)}. \tag{8}$$

Now, as U' is a proper subset of V_1, and the weights are positive,

$$\text{ratioCut}(U') < \text{ratioCut}(V_1).$$

□

In fact, we can establish even a stronger property about the ratioCuts and the subsets in the Principal Partition.

Lemma 12 *If $X_1 \subset X_2$ are both in the Principal Partition of $(w_e(\delta(.)), w_v(.))$ for $V_1 \subset V$, then $ratioCut(X_1) < ratioCut(X_2)$.*

(Proof of Lemma 12 :)
Clearly it suffices to prove that $\frac{w_e(\delta(X_1))}{w_v(X_1)} \leq \frac{w_e(\delta(X_2))}{w_v(X_2)}$. This is indeed implied by a more general result in [16]. But for the sake of completeness of the presentation, we provide a direct proof.

Let λ_1, λ_2 be such that

$$w_e(\delta(X_1)) - \lambda_1 * w_v(X_1) = \min_{U \subseteq V_1}(w_e(\delta(U)) - \lambda_1 * w_v(U))$$

and

$$w_e(\delta(X_2)) - \lambda_2 * w_v(X_2) = \min_{U \subseteq V_1}(w_e(\delta(U)) - \lambda_2 * w_v(U)).$$

As $k_1 < k_2$ implies $X^{k_1} \subseteq X_{k_2}$, $X_1 \subset X_2$ implies $\lambda_1 \leq \lambda_2$. Let $g(.)$ denote $w_e(\delta(.))$.

We have, $g(X_1) - \lambda_1 * w_v(X_1) \leq g(X_2) - \lambda_1 * w_v(X_2)$ and $g(X_2) - \lambda_2 * w_v(X_2) \leq g(X_1) - \lambda_2 * w_v(X_1)$. The above implies

$$\lambda_1 \leq \frac{g(X_2) - g(X_1)}{w_v(X_2) - w_v(X_1)} \leq \lambda_2.$$

We also note that $g(X_1) - \lambda_1 * w_v(X_1) \leq g(\emptyset) - \lambda_1 * w_v(\emptyset)$, implying $\frac{g(X_1)}{w_v(X_1)} \leq \lambda_1$. Let α denote $\frac{g(X_1)}{w_v(X_1)}$. Thus $\alpha \leq \lambda_1$.

Now to show that $ratioCut(X_1) < ratioCut(X_2)$, it suffices to show that $\frac{g(X_2)}{w_v(X_2)} \geq \alpha$, as $w_v(V - X_1) > w_v(V - X_2)$. Suppose the contrary, i.e. $\frac{g(X_2)}{w_v(X_2)} < \alpha$. This would mean

$$\lambda_1 \leq \frac{g(X_2) - g(X_1)}{w_v(X_2) - w_v(X_1)} < \alpha,$$

a contradiction. $\square$

The following lemma shows that we could get a better ratioCut if we use maximum flow-minimum cut to separate a pair of critical sets in the Principal Sequences associated with V_1 and its complement, i.e. V_2.

Lemma 13 *Let $G(V, E, w_v, w_e)$ be a given graph and let (V_1, V_2) be a bipartition of V. Let U_1 be a nonempty subset in the Principal Sequence of $w_e(\delta(.))$ for $G(V, E, w_e, w_v)$ and subset V_1. Let U_2 be a nonempty subset in the Principal Sequence of $w_e(\delta(.))$ for $G(V, E, w_e, w_v)$ and subset $V_2 = V - V_1$.*

Let $(X, V - X)$ be a minimum-cut separating U_1 and U_2, with $U_1 \subseteq X$ and $U_2 \subseteq V - X$. Then,

$$\begin{aligned}\text{ratioCut}(X) &\leq \max\{\text{ratioCut}(U_1), \text{ratioCut}(U_2)\} \\ &\leq \text{ratioCut}(V_1).\end{aligned}$$

The inequality in the above is strict if $X \neq U_1$ and $V - X \neq U_2$.

(Proof of Lemma 13 :)
Observe that

$$w_e(\delta(X)) \leq \min\{w_e(\delta(U_1)), w_e(\delta(U_2))\}. \tag{9}$$

First we consider the case that $w_v(U_1) \leq w_v(U_2)$. Since $(X, V - X)$ is a cut separating U_1 and U_2,

$$w_v(X) \geq w_v(U_1) \tag{10}$$

and

$$w_v(V - X) \geq w_v(U_2). \tag{11}$$

The inequality 11 implies $w_v(X) \leq w_v(V - U_2) \leq w_v(V - U_1)$. The last inequality in the above follows from our assumption in this case that $w_v(U_1) \leq w_v(U_2)$.

Thus $w_v(U_1) \leq w_v(X) \leq w_v(V - U_1)$. From this it is easy to see that $w_v(U_1) * w_v(V - U_1) \leq w_v(X) * w_v(V - X)$. Now from the above observation and inequality 9 it follows that $ratioCut(X) \leq ratioCut(U_1)$.

Considering the symmetrical case of $w_v(U_2) \leq w_v(U_1)$ in a similar manner, the assertion in the lemma follows. $\square$

It is clear that if there exists a nonempty subset $W \subseteq V_1$ satisfying $\frac{w_e(\delta(W))}{w_v(W)} < \frac{w_e(\delta(V_1))}{w_v(V_1)}$, then $ratioCut(W) < ratioCut(V_1)$, i.e. there is scope of improving the ratioCut. We note using the following that our algorithm would indeed in such a case find an improved bipartition.

Lemma 14 *Let $G(V, E, w_v, w_e)$ be a given graph and let (V_1, V_2) be a bipartition of V. Suppose there exists a nonempty subset W of V_1 such that*

$$\frac{w_e(\delta(W))}{w_v(W)} < \frac{w_e(\delta(V_1))}{w_v(V_1)}. \tag{12}$$

Further let

$$\hat{\lambda} = \frac{w_e(\delta(V_1))}{w_v(V_1)}. \tag{13}$$

Then the largest subset, say Z, of V_1 that solves

$$\min_{U \subseteq V_1} (w_e(\delta(U)) - \hat{\lambda} * w_v(U)) \tag{14}$$

is such that

$$\text{ratioCut}(Z) < \text{ratioCut}(V_1).$$

(Proof of Lemma 14 :)
Clearly, $w_e(\delta(W)) - \hat{\lambda} * w_v(W) < 0$. Thus neither $\emptyset$ nor V_1 minimizes $w_e(\delta(.)) - \hat{\lambda} * w_v(.)$ over the subsets of V_1, implying there is a nonempty proper subset of V_1 in the Principal Sequence. So the largest subset Z that minimizes $w_e(\delta(.)) - \hat{\lambda} * w_v(.)$ over the subsets of V_1 is a nonempty and proper subset of V_1. The assertion now follows from Lemma 11. $\square$

It is clear from the above proof that, if we want to find an improved bipartition without computing the whole Principal Sequence, then we could make use of the above $\hat{\lambda}$. Indeed, if there exists a nonempty proper subset of V_1 in the Principal Sequence considered here, one such subset is the largest subset that solves $\min_{U \subseteq V_1} (w_e(\delta(U)) - \hat{\lambda} * w_v(U))$.

For the practical utility of this observation in view of inaccuracy in floating point arithmetic, we shall show in the next lemma that our above idea of using $\hat{\lambda} = \frac{w_e(\delta(V_1))}{w_v(V_1)}$ is robust.

Lemma 15 *Let $G(V, E, w_v, w_e)$ be a given graph and let (V_1, V_2) be a bipartition of V. Suppose there exists a nonempty subset W of V_1 such that*

$$\frac{w_e(\delta(W))}{w_v(W)} < \frac{w_e(\delta(V_1))}{w_v(V_1)}. \tag{15}$$

Then there exists a (nonempty) open interval (λ', λ'') such that the largest subset, say Z, of V_1 that solves

$$\min_{U \subseteq V_1} (w_e(\delta(U)) - \lambda * w_v(U)) \tag{16}$$

for any $\lambda \in (\lambda', \lambda'')$ is such that

$$\text{ratioCut}(Z) < \text{ratioCut}(V_1).$$

(Proof of Lemma 15 :)
Choose $\hat{\lambda} = \frac{w_e(\delta(V_1))}{w_v(V_1)}$. Clearly, $w_e(\delta(W)) - \hat{\lambda} * w_v(W) < 0$. Thus neither $\emptyset$ nor V_1 minimizes $w_e(\delta(.)) - \hat{\lambda} * w_v(.)$ over the subsets of V_1, implying there is a nonempty proper subset of V_1 in the Principal Sequence. Thus the Principal Sequence consists of at least 3 critical sets, therefore the smallest and largest critical values, say λ' and λ'' are distinct. From the properties of the Principal Sequence mentioned in an earlier section, for any $\lambda \in (\lambda', \lambda'')$, X^{λ} is a nonempty and proper subset of V_1. The rest follows from Lemma 11. $\square$

5 The Case of Failure to Find an Improved Bipartition

5.1 A Performance Guarantee

We have already noted in the earlier sections (see Lemma 11 and discussion around it!) that presence of a nonempty proper subset of V_1 in the Principal Sequence of $w_e(\delta(.))$ for $G(V, E, w_v, w_e)$ and $V_1 \subseteq V$, would mean that we can use such a set to improve the ratioCut.

On the other hand, absence of any nonempty proper subset of V_1 in the Principal Sequence of $w_e(\delta(.))$ for $G(V, E, w_v, w_e)$ and $V_1 \subseteq V$ also offers the following performance guarantee.

Proposition 16 *Let $G(V, E, w_v, w_e)$ be a given connected simple graph with positive edge weights given by $w_e(.)$ and nonnegative node weights given by $w_v(.)$. Let $(V_1, V - V_1)$ be a given bipartition of V satisfying $w_v(V_1) \neq 0$ and $w_v(V - V_1) \neq 0$.*

If there is no nonempty and proper subset of V_1 in the Principal Sequence of $w_e(\delta(.))$ for $G(V, E, w_v, w_e)$ and $V_1 \subseteq V$, then

$$ratioCut(V_1) \leq \alpha * \min_{\emptyset \neq W \subseteq V_1} ratioCut(W)$$

where $\alpha = \frac{w_v(V)}{w_v(V-V_1)}$.

(Proof of Proposition 16:)
Let $\hat{\lambda} = \frac{w_e(\delta(V_1))}{w_v(V_1)}$.

From the hypothesis of this proposition that there is no nonempty and proper subset of V_1 in the Principal Sequence of $w_e(\delta(.))$ for $G(V, E, w_v, w_e)$ and $V_1 \subseteq V$, it follows that

$$\min_{W \subseteq V_1} (w_e(\delta(W)) - \hat{\lambda} * w_v(W)) = 0. \tag{17}$$

For otherwise, $\min_{W \subseteq V_1}(w_e(\delta(W)) - \hat{\lambda} * w_v(W)) \neq 0$. And as $(w_e(\delta(V_1)) - \hat{\lambda} * w_v(V_1)) = (w_e(\delta(\emptyset)) - \hat{\lambda} * w_v(\emptyset)) = 0$, it clearly implies that neither $\emptyset$ nor V_1 solve $\min_{W \subseteq V_1}(w_e(\delta(W)) - \hat{\lambda} * w_v(W))$. This clearly implies that there exists a nonempty and proper subset of V_1 in the Principal Sequence of $w_e(\delta(.))$ for $G(V, E, w_v, w_e)$ and $V_1 \subseteq V$, which contradicts the hypothesis.

Now for $\emptyset \neq W \subseteq V_1$,

$$ratioCut(W) = \frac{w_e(\delta(W))}{w_v(W) * w_v(V - W)} \geq \frac{\hat{\lambda}}{w_v(V - W)}.$$

The above follows as $w_e(\delta(W)) - \hat{\lambda} * w_v(W) \geq 0$ (which itself follows from equation 17), which implies that $\frac{w_e(\delta(W))}{w_v(W)} \geq \hat{\lambda}$. As $w_v(V - W) \leq w_v(V)$, we get

$$ratioCut(W) \geq \frac{\hat{\lambda}}{w_v(V)}.$$

Using the following trivial identity

$$ratioCut(V_1) = \frac{\hat{\lambda}}{w_v(V - V_1)}$$

that follows from the definition of $\hat{\lambda}$, we get

$$ratioCut(W) \geq ratioCut(V_1) * \frac{w_v(V - V_1)}{w_v(V)}.$$

Choice of $\alpha = \frac{w_v(V)}{w_v(V-V_1)}$ proves the proposition. $\square$

The above performance guarantee is significant as the following section shows the related problem is NP-hard.

5.2 NP-hardness of Constrained Weighted Sparsest Cut Problem

Constrained Weighted Sparsest Cut Problem is stated as follows:
Let $G(V, E, w_v, w_e)$ be a given graph with $w_v(.) > 0$ and $w_e(.) > 0$. Furthermore, we assume that the graph is connected (i.e. $w_e(\delta(P)) > 0 \; \forall \emptyset \neq P \subseteq V$). Let $\emptyset \neq U \subset V$ be a specified vertex subset. Solve

$$\min_{\emptyset \neq W \subseteq U} ratioCut(W).$$

We show that this problem is NP-hard by reducing the known NP-hard problem of computing **Sparsest Cut** [14], that is, solving

$$\min_{\emptyset \neq W \subseteq V'} ratioCut(W)$$

in a connected simple graph (i.e. a graph without loops and parallel edges) $G'(V', E')$ with unit edge and node weights.

We reduce the above **Sparsest Cut Problem** to the **Constrained Weighted Sparsest Cut Problem**. Let $G(V, E, w_v, w_e)$ be a given graph with $w_v(.) = 1$ and $w_e(.) = 1$ for all vertices and edges respectively. Furthermore, we assume that the graph is connected.

Let $G'(V', E', w'_v, w'_e)$ be obtained from the above graph by adding a complete subgraph on $V' - V$ in which each edge has infinite weight and one of the nodes in $V' - V$ is connected to one of the nodes in V by an infinite weight edge. Each node in $V' - V$ has appropriately small weight.

It is easy to see that, by solving **Constrained Weighted Sparsest Cut Problem** on this augmented graph while specifying the subset V of V', we get a solution to **Sparsest Cut Problem**.

As the above reduction is clearly a polynomial time reduction, **Constrained Weighted Sparsest Cut Problem** is NP-hard.

6 Computational Results

We tested part of our approach on benchmark circuits. We used a subset of ACM/SIGDA benchmarks (available via World Wide Web at http://ballade.cs.ucla.edu/cheese). We obtained a good initial bipartition using the standard Fiduccia-Matheyses (FM) method. For each block of the bipartition, we attempted to compute only one set in the Principal Sequence (as described in Lemma 14).

Circuit	FM-RatioCut	RatioCut1	RatioCut2
prim1	6.062	1.734	2.645
prim1	4.733	2.825	3.853
prim1	5.852	5.394	2.517
prim2	1.002	0.397	
test02	2.834	1.651	2.243
test02	2.836	1.839	2.660
test02	2.352	0.999	2.049
test03	2.731	1.166	1.455
test03	2.543	1.352	2.162
test03	1.737	1.528	1.321
test03	2.666	1.724	1.915
test03	1.750	1.353	1.237
test04	3.400	2.429	1.411
test04	2.316	1.819	1.112
test04	2.707	1.799	1.759
test04	2.177	1.564	1.315
test04	1.576	0.804	1.118
test05	1.431	0.817	1.122
test05	1.515	0.647	0.771
test05	1.319	0.842	1.094

Note: The ratioCuts are scaled up by 10^4.

The above table shows that even this was consistently enough to produce significantly better bipartitions in the sense of the ratioCut. The column *FM-RatioCut* shows the ratioCut of the bipartition found by the FM method. The columns *RatioCut1* and *RatioCut2* show the values of the ratioCuts obtained by our algorithm described in section 4.2 which attempts to find a set, each in the Principal Partition of the $(w_e(\delta(.)), w_v(.))$ on V_1 and V_2 respectively.

7 Application of Submodular Functions to Weakest Linked k-Vertex Subset Problem

In this section we describe a new polynomial time approach that finds 2-approximate solutions to the Weakest Linked k-Vertex Subset Problem for the given unweighted graph $G(V, E)$. We recall that the problem *weakest linked k-vertex subset* for a graph $G(V, E)$, i.e. solving $\min_{U \subseteq V, |U|=k} |\Gamma(U)|$ for a specified k, is NP-hard.

Given an unweighted and undirected graph $G(V, E)$, we consider the Principal Partition of $|\Gamma(.)|$ ($|\Gamma(.)|$ is a submodular function). The *Principal Partition* of $\Gamma(.)$ is the collection of subsets that solve

$$\min_{U \subseteq V} (|\Gamma(U)| - \lambda * |U|)$$

for some real λ.

From the theory of Principal Partition of submodular function [20] (see also for details [8,16]), it is clear that, $\exists$ a unique nested sequence

$$\emptyset = U_0 \subset U_1 \subset U_2 \ldots \subset U_t = V$$

and an associated unique sequence of reals (called critical values)

$$\lambda_1 < \lambda_2 < \lambda_3 \ldots < \lambda_t$$

satisfying U_{i-1} and U_i are smallest and largest (respectively) subsets that solve

$$\min_{U \subseteq V} (|\Gamma(U)| - \lambda_i * |U|).$$

Once again using the generic algorithm to compute the Principal Sequence of a given submodular function [16], we can compute the above nested sequence of subsets of vertices, *i.e.*

$$\emptyset = U_0 \subset U_1 \subset U_2 \ldots \subset U_t = V$$

by $O(|V|)$ invocations of the subroutine that computes smallest or largest subset that solves

$$\min_{U \subseteq V}(|\Gamma(U)| - \lambda * |U|).$$

It is easy to see that a network flow approach (similar to ones used by [9]) can be used to develop the above mentioned subroutine. Thus the time complexity of the minimization subroutine is equivalent to that of a single invocation of that of computing maximum flow in a flow network of size same as that of $G(V, E)$ (see [11]).

7.1 An Approximation Algorithm

The following straightforward result shows that Principal Partition provides **optimal** solutions to some instances of this NP-hard problem.

Proposition 17 *Let $\hat{U}$ be in the Principal Partition of $|\Gamma(.)|$. Then $\hat{U}$ is an optimal solution to* **weakest linked k-vertex subset** *problem for $k = |\hat{U}|$.*

(Proof of Proposition 17:)
Let λ be such that

$$|\Gamma(\hat{U})| - \lambda * |\hat{U}| = \min_{U \subseteq V}(|\Gamma(U)| - \lambda * |U|).$$

This implies

$$|\Gamma(\hat{U})| - \lambda * |\hat{U}| = \min_{U \subseteq V, |U| = |\hat{U}|}(|\Gamma(U)| - \lambda * |U|).$$

Thus

$$|\Gamma(\hat{U})| = \min_{U \subseteq V, |U| = k} |\Gamma(U)|.$$

□

Next we show how an approximate solution could be found to the problem of *weakest linked k-vertex subset* for an arbitrary integer k.

Let k be the given integer. Let $U_0 \subset U_1 \subset U_2 \ldots \subset U_t$ and $\lambda_1 < \lambda_2 < \lambda_3 \ldots < \lambda_t$ be as above.

Let $|U_{p-1}| \leq k < |U_p|$. In the case where $|U_{p-1}| = k$, we know from Proposition 17 that U_{p-1} is indeed an optimal solution.

Consider the case $|U_{p-1}| < k < |U_p|$.

Let $\{\alpha_1, \alpha_2, \ldots, \alpha_q\} = U_p - U_{p-1}$ be assumed to be ordered so that

$$|\delta(\alpha_j) \cap E(V - U_{p-1})| \leq |\delta(\alpha_{j+1}) \cap E(V - U_{p-1})|$$

for $j = 1, 2, \ldots, q - 1$.

$$\begin{aligned}
&\sum_{j=1}^{k-|U_{p-1}|} |\delta(\alpha_j) \cap E(V - U_{p-1})| \\
\leq& \frac{k - |U_{p-1}|}{|U_p| - |U_{p-1}|} * \sum_{j=1}^{|U_p|-|U_{p-1}|} |\delta(\alpha_j) \cap E(V - U_{p-1})| \\
\leq& \frac{k - |U_{p-1}|}{|U_p| - |U_{p-1}|} * 2 * (|\Gamma(U_p)| - |\Gamma(U_{p-1})|) \\
\leq& 2 * \lambda_p * (k - |U_{p-1}|).
\end{aligned}$$

But

$$\begin{aligned}
&|\Gamma(U_{p-1} \cup \{\alpha_1, \alpha_2 \ldots, \alpha_{k-|U_{p-1}|}\})| \\
\leq& |\Gamma(U_{p-1})| + \textstyle\sum_{j=1}^{k-|U_{p-1}|} |\delta(\alpha_j) \cap E(V - U_{p-1})| \\
\leq& |\Gamma(U_{p-1})| + 2 * \lambda_p * (k - |U_{p-1}|).
\end{aligned} \tag{18}$$

However, from the properties of Principal Partition it is easy to see that (see [16]),

Proposition 18 *Let k be the given integer and let $U_0 \subset U_1 \subset U_2 \ldots \subset U_t$ and $\lambda_1 < \lambda_2 < \lambda_3 < \ldots < \lambda_t$ be as described above. Then*

$$\min_{U \subseteq V, |U|=k} |\Gamma(U)| \geq |\Gamma(U_{p-1})| + \lambda_p * (k - |U_{p-1}|).$$

(Proof of Proposition 18:)
For any $U \subseteq V$ with $|U| = k$,

$$|\Gamma(U)| - \lambda_p * k = |\Gamma(U)| - \lambda_p * |U| \geq |\Gamma(U_{p-1})| - \lambda_p * |U_{p-1}|.$$

That is, for any subset $U \subseteq V$ with $|U| = k$,

$$|\Gamma(U)| \geq |\Gamma(U_{p-1})| + \lambda_p * (k - |U_{p-1}|).$$

□

Thus from the relation 18 and the above Proposition 18, the following theorem is obtained.

Theorem 19 $U_{p-1} \cup \{\alpha_1, \alpha_2 \ldots, \alpha_{k-|U_{p-1}|}\}$ *as above, is a 2-approximate solution to the* weakest linked k-vertex subset *problem.*

8 Conclusion

We have shown the applications of submodular functions to some NP-hard problems arising in the context of VLSI Design Automation. Our algorithms are based on the theory of submodular functions. They possess strong combinatorial flavor and are efficiently implementable using network flow based subroutines.

Acknowledgements

The authors thank the anonymous referees for their comments and suggestions.

References

[1] Y. Asahiro, K. Iwama and H. Tamaki : Greedily finding a dense subgraph, *Journal of Algorithms*, vol. 34, 2000, pp. 203-221.

[2] C.J. Alpert, A.B.Kahng and D.S. Yao : Spectral Partitioning with Multiple Eigenvectors, *Discrete Applied Mathematics*, vol. 90, 1999, pp. 3-26.

[3] C.J. Alpert and Yao : Spectral Partitioning: The More Eigenvectors, The Better, *Proceedings of 32nd Design Automation Conference*, 1995, pp.195-200.

[4] E. Cheng and W. Cunningham : A faster algorithm for computing the strength of a network, *Information Processing Letters*, vol. 49, 1994, pp.209-212.

[5] W. Dinkelbach : On nonlinear fractional programming, *Management Sci.*, vol. 13, 1967, pp.492-498.

[6] J. Edmonds : Submodular functions, matroids and certain polyhedra, *Proc. Calgary Intl. conference on Combinatorial Structures*, 1970, pp.69-87.

[7] U. Feige, G . Kortsarz and D. Peleg : The Densest k-Subgraph Problem, *Proceedings of the* 34^{th} *ann. Symposium on Foundations of Computer Science*, 1993, pp. 692-701.

[8] S. Fujishige : *Submodular Functions and Optimization, Annals of Discrete Mathematics*, North Holland, 1991.

[9] G. Gallo, M. Grigoriadis and R.E. Tarjan : A fast parametric network flow algorithm, *SIAM J. of Computing*, vol. 18, 1989, pp.30-55.

[10] H.N. Gabow : Algorithms for graphic polymatroids and parametric s-sets *J. Algorithms*, vol. 26, 1998, pp.48-86.

[11] A.V. Goldberg and R.E. Tarjan : A new approach to the maximum flow problem, *J. Assoc. Computing Mach.*, vol. 35, 1988.

[12] O. Goldschmidt and D. Hochbaum : Polynomial algorithm for the k-cut problem, in 29^{th} *ann. Symposium on Foundations of Computer Science*, 1988, pp. 444-451.

[13] D. Gusfield : Computing the strength of a graph, *SIAM J. of Computing*, vol. 20, 1991, pp.639-654.

[14] F.T. Leighton and S. Rao : An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximate algorithms, 29^{th} *ann. Symposium on Foundations of Computer Science*, 1988, pp. 422-431.

[15] T. Lengauer : *Combinatorial Algorithms for Integrated Circuit Layout*, Bryan Press, New York, 1990.

[16] H. Narayanan : *Submodular Functions and Electrical Networks, Annals of Discrete Mathematics-54*, North Holland, 1997.

[17] H. Narayanan, S. Roy and S.B. Patkar : Approximation Algorithms for min-k-overlap Problems, using the Principal Lattice of Partitions Approach, *J. of Algorithms*, vol. 21, 1996, pp. 306-330.

[18] S.B. Patkar and H. Narayanan : Fast On-line/Off-line Algorithms for Optimal Reinforcement of a Network and its Connections with Principal Partition, *Proceedings of* 20^{th} *ann. Foundations of Software Technology and Theoretical Computer Science, LNCS-1974, Springer*, 2000, pp. 94-105.

[19] J.C. Picard and M. Queyeranne : Selected applications of Minimum Cuts in Networks, *INFOR-Canada Journal of Operations Research Inform. Process.*, vol. 20, 1982, pp.394-422.

[20] N. Tomizawa and S. Fujishige : Historical survey of extensions of the concept of Principal Partition and their unifying generalization to hypermatroids. Systems Science Research Report No.5, Department of Systems Science, Tokyo Institute of Technology, April 1982 (also its abridgment in *Proceedings of the 1982 IEEE International Symposium on Circuits and Systems* Rome, 1982, pp. 142-145).