In Combinatorica, 15 (1995).

Minimizing Submodular Functions over Families of Sets*

M.X. Goemans† V.S. Ramakrishnan‡

December 1993
Revised: July 1995

Abstract

We consider the problem of characterizing the minimum of a submodular function when the minimization is restricted to a family of subsets. We show that, for many interesting cases, there exist two elements a and b of the groundset such that the problem is equivalent to the problem of minimizing the submodular function over the sets containing a but not b. This leads to a polynomial-time algorithm for minimizing a submodular function over these families of sets. Our results apply, for example, to the families of odd cardinality subsets or even cardinality subsets separating two given vertices, or to the complement of a lattice family of subsets. We also derive that the second smallest value of a submodular function over a lattice family can be computed in polynomial-time. These results generalize and unify several known results.

1 Introduction

Submodular set-functions arise in a variety of fields, including combinatorial optimization, probability and geometry. Examples include the rank function of a matroid, the sizes of cutsets in a directed or undirected graph, the probability that a subset of events do not simultaneously occur, or the logarithm of the volume of the parallelepiped spanned by a subset of linearly independent vectors. For a survey of submodular functions and their properties, the reader is referred to Lovász [9] and Fujishige [2].

Many problems in combinatorial optimization can be formulated as the problem of minimizing a submodular function. Illustrations include the minimum cut or s-t cut proBlem in an undirected or directed graph, or the problem of finding the largest independent set common to two matroids. In several settings though, the minimization of the submodular

*Research partially supported by NSF contract 9302476-CCR, Air Force contract F49620-92-J-0125 and DARPA contract N00014-92-J-1799.

†Department of Mathematics, Room 2-382, MIT, Cambridge, MA 02139. goemans@math.mit.edu.

‡McKinsey & Co., Boston, MA.

function is restricted to a family of subsets. In the minimum cut problem, one would like to exclude the empty set or the entire set, or in the minimum odd cut problem, one would like to consider only the subsets with odd cardinality. Several restricted minimum cut problems have been shown to be polynomially solvable: the odd-cut or T-odd cut problem by Padberg and Rao [12], the even-cut or T-even cut problem by Barahona and Conforti [1] and any "proper" cut problem by Gabow, Goemans and Williamson [3]. Examples of families of proper cuts are the T-odd cuts and the generalized Steiner cuts. Grötschel, Lovász and Schrijver [6] (see also [7, 8]) have generalized Padberg and Rao's result to any submodular function and to more general families of subsets. Their algorithm can be used to solve the minimum s-t T-odd cut or s-t T-even cut in polynomial time.

In this paper, we consider what we call *parity* families, which are even more general families of subsets than the ones considered in Grötschel et al. [8]. We show that the minimization of a submodular function over a parity family can be done in polynomial time. Our algorithm and the corresponding characterization differs significantly from the approach used in Grötschel et al. [8]. Their algorithm as well as the original algorithm of Padberg and Rao and the algorithm of Barahona and Conforti exploit certain uncrossing properties of minimum sets or minimum cuts. We show that, in general, this property does not hold for parity families, rendering the approach infeasible. Our characterization, however, shows the existence of two elements a and b (or two vertices) such that the minimization over the parity family is equivalent to the minimization over the sets containing a but not b.

The paper is structured as follows. In Section 2, we define all the necessary terminology and we describe our main result and its algorithmic implications. Section 3 describes special cases of our result and also relates our result to results from the literature. The proof of our main result is given in Section 4 and the derivation of the algorithmic consequences is discussed in Section 5. In the last section, we use the techniques developed in this paper to give a simple proof of a recent result of Nagamochi et al. [11] on the maximum number of approximately minimum cuts in an undirected graph.

2 Definitions, Preliminaries and Results

Let V be a finite set. A family $\mathcal{C}$ of subsets of V is called a *lattice family* if

$$A, B \in \mathcal{C} \Rightarrow A \cap B, A \cup B \in \mathcal{C}.$$

We also introduce a new type of family of subsets. A subfamily $\mathcal{G}$ of a lattice family $\mathcal{C} \subseteq 2^V$ is called a *parity family*[1] if

$$A, B \in \mathcal{C} \setminus \mathcal{G} \Rightarrow (A \cap B \in \mathcal{G} \text{ iff } A \cup B \in \mathcal{G}).$$

[1] Often, to emphasize the dependence on $\mathcal{C}$, we shall refer to $\mathcal{G}$ as a parity subfamily of $\mathcal{C}$.

A related family of subsets was introduced by Grötschel, Lovász and Schrijver [7] to provide a general setting for the minimization of a submodular function over odd-cardinality or even-cardinality members of a lattice family. A subfamily $\mathcal{G} \subseteq \mathcal{C}$ is a *triple* family if, for any $S, T \in \mathcal{C}$, whenever three of the four sets S, T, $S \cap T$ and $S \cup T$ are in $\mathcal{C} \setminus \mathcal{G}$, then the fourth set is also in $\mathcal{C} \setminus \mathcal{G}$. It is easy to see that triple families are a subclass of parity families. We will give several examples of triple and parity families in Section 3. We will also explore further the relationship between these families and the differences in algorithmic approaches to minimizing submodular functions over them.

Given a family $\mathcal{C} \subseteq 2^V$ of subsets and two elements s and t in V, we define $\mathcal{C}_{st}$ to be the family $\{A \in \mathcal{C} : s \in A, t \notin A\}$. If $\mathcal{C}$ is a lattice family then $\mathcal{C}_{st}$ is still a lattice family. Also, if $\mathcal{G}$ is a parity subfamily of $\mathcal{C}$ then $\mathcal{G}_{st}$ is a parity subfamily[2] of $\mathcal{C}_{st}$.

A function $f : 2^V \rightarrow \mathbb{R}$ is said to be *submodular on* 2^V if

$$f(A) + f(B) \geq f(A \cap B) + f(A \cup B) \qquad \text{for all } A, B \subseteq V.$$

A function f is said to be *submodular on the family* $\mathcal{C}$ of subsets if

$$f(A) + f(B) \geq f(A \cap B) + f(A \cup B) \qquad \text{whenever } A, B, A \cup B, A \cap B \in \mathcal{C}.$$

We say that a set S minimizes a function f over a family $\mathcal{C}$ if $S \in \mathcal{C}$ and $f(S)$ is as small as possible. Throughout, we assume that the submodular function f is given by an oracle returning $f(S)$ for each query $S \subseteq V$. Minimizing a submodular function over a lattice family can be done in oracle-polynomial time[3] via the ellipsoid algorithm, as was shown by Grötschel, Lovász and Schrijver [6]. This also trivially implies that one can minimize a submodular function over the union of a polynomial number of lattice families in oracle-polynomial time. For example, given a set $T \subseteq V$, one can minimize a submodular function over $\mathcal{F} = \{S : S \cap T \neq \emptyset, T \setminus S \neq \emptyset\}$ by expressing the family as $\mathcal{F} = \bigcup_{s,t \in T, s \neq t} 2^V_{st}$. In this case, the number of submodular function minimizations over lattice families can be reduced to $2|T| - 2$ by expressing $\mathcal{F}$ as

$$\mathcal{F} = \bigcup_{t \in T \setminus \{s\}} (2^V_{st} \cup 2^V_{ts}) \tag{1}$$

for any element $s \in T$, or even to only $|T|$ submodular function minimizations since

$$\mathcal{F} = \bigcup_{i=1}^{l} 2^V_{t_i t_{i+1}} \tag{2}$$

where $T = \{t_1, \ldots, t_l\}$ and $t_{l+1} = t_1$.

[2] It is important to embed $\mathcal{G}_{st}$ in $\mathcal{C}_{st}$; $\mathcal{G}_{st}$ may not be a parity subfamily of $\mathcal{C}$.

[3] Oracle-polynomial time means that the number of operations, counting each call to the oracle as one operation, is polynomially bounded in $|V|$ and in an upper bound β on the size of $f(S)$ for any S.

Throughout the paper, *minimal* and *maximal* refer to minimal and maximal *under inclusion.*

The following theorem constitutes our main result.

Theorem 1 *Let $\mathcal{C} \subseteq 2^V$ be a lattice family and $\mathcal{G} \subseteq \mathcal{C}$ be a parity family. Let f be a submodular function on $\mathcal{C}$. Let $S^* \in \mathcal{G}$ be a set minimizing f over $\mathcal{G}$. Then either $S^* \in \{\emptyset, V\}$ or there exist $a, b \in V$ such that S^* minimizes f over the lattice family $\mathcal{C}_{ab}$.*

The previous theorem is however not sufficient to derive a polynomial-time algorithm for the minimization over parity families. Indeed, even if we know a and b, the minimization over the lattice family $\mathcal{C}_{ab}$ might not return a set belonging to $\mathcal{G}$. Nevertheless, Theorem 1 can be refined as follows.

Theorem 2 *Let $\mathcal{C} \subseteq 2^V$ be a lattice family and $\mathcal{G} \subseteq \mathcal{C}$ be a parity family. Let f be a submodular function on $\mathcal{C}$. Let $S^* \in \mathcal{G}$ be a* minimal *set minimizing f over $\mathcal{G}$. Then either $S^* \in \{\emptyset, V\}$ or there exist $a, b \in V$ such that S^* is the* unique minimal *set minimizing f over the lattice family $\mathcal{C}_{ab}$.*

Theorem 2 implies the following algorithmic consequence.

Corollary 3 *Let $\mathcal{C} \subseteq 2^V$ be a lattice family and $\mathcal{G} \subseteq \mathcal{C}$ be a parity family. Let f be a submodular function on $\mathcal{C}$. Then a set minimizing f over $\mathcal{G}$ can be obtained in oracle-polynomial time by solving $O(|V|^2)$ submodular function minimizations over lattice families.*

In the next section, we elaborate on applications and special cases of our main result and its algorithmic implication.

3 Previous Work and Applications

We begin by reviewing some examples of triple families.

Lemma 4 (Grötschel et al. [8]) *Let $\mathcal{C} \subseteq 2^V$ be any lattice family and T be a non-empty subset of V. Then, for any $p, q \in \mathbb{Z}$, the family $\mathcal{G} := \{S \in \mathcal{C} : |S \cap T| \not\equiv q \pmod{p}\}$ is a triple family.*

In particular, letting $p = 2$, we conclude that both the family of odd sets and the family of even sets are triple families. More generally, given integers a_i for $i \in V$ and integers p and q, the family $\mathcal{G} = \{S \in \mathcal{C} : \sum_{i \in S} a_i \not\equiv q \pmod{p}\}$ is a triple family. Lemma 4 corresponds to the case $a_i = 1$ for $i \in T$ and 0 otherwise. The triple family $\{S \in \mathcal{C} : |S \cap T_1| \not\equiv |S \cap T_2| \pmod{p}\}$ corresponds to the case $a_i = 1$ for $i \in T_1 \setminus T_2$, $a_i = -1$ for $i \in T_2 \setminus T_1$, $a_i = 0$ otherwise, and $q = 0$.

Grötschel et al. [8] also show that $\mathcal{C} \setminus \mathcal{A}$ is a triple family where $\mathcal{C}$ is a lattice family and $\mathcal{A}$ is an antichain. A family $\mathcal{A}$ is called an *antichain* if, for distinct $S_1, S_2 \in \mathcal{A}$, $S_1 \not\subseteq S_2$. For example, the exclusion of all sets of cardinality q leads to a triple family (this also follows from Lemma 4 with $p > |V|$ and $T = V$).

We observed earlier that triple families are a subclass of parity families and, thus, the above families are also parity families. However, the inclusion is strict. An important class of parity families which are not necessarily triple families is given in the following lemma.

Lemma 5 *Let $\mathcal{F} \subseteq \mathcal{C} \subseteq 2^V$ where $\mathcal{F}$ and $\mathcal{C}$ are both lattice families. Then $\mathcal{G} = \mathcal{C} \setminus \mathcal{F}$ is a parity family.*

The proof is immediate. Since $\mathcal{C} \setminus \mathcal{G} = \mathcal{F}$ is a lattice family, the condition $A, B \in \mathcal{C} \setminus \mathcal{G}$ implies that both $A \cap B$ and $A \cup B$ also belong to $\mathcal{C} \setminus \mathcal{G}$.

To argue that the families addressed in Lemma 5 are typically not triple families, consider the special case where $\mathcal{F}$ is a chain. A family $\mathcal{F}$ of subsets is a *chain* if, for any $S_1, S_2 \in \mathcal{F}$, $S_1 \subseteq S_2$ or $S_2 \subseteq S_1$. If $\mathcal{F}$ is a chain with $|\mathcal{F}| \geq 3$ then $\mathcal{C} \setminus \mathcal{F}$ is *not* a triple family. To show this, let S_1, S_2 and S_3 be distinct members of $\mathcal{F}$. Since $\mathcal{F}$ is a chain, we may assume that $S_1 \subset S_2 \subset S_3$. Consider the subsets S_2 and $(S_3 \setminus S_2) \cup S_1$. Now $S_2 \in \mathcal{F}$, $S_2 \cap ((S_3 \setminus S_2) \cup S_1) = S_1 \in \mathcal{F}$ and $S_2 \cup ((S_3 \setminus S_2) \cup S_1) = S_3 \in \mathcal{F}$ but $(S_3 \setminus S_2) \cup S_1 \in \mathcal{C} \setminus \mathcal{F}$. Therefore $\mathcal{C} \setminus \mathcal{F}$ cannot be a triple family.

If f is a submodular function on a lattice family $\mathcal{C}$ then it is easy to see that $\mathcal{F}_{min} = \{A \in \mathcal{C} : f(A) = \min_{S \in \mathcal{C}} f(S)\}$ is a lattice family. Lemma 5 then guarantees that $\mathcal{C} \setminus \mathcal{F}_{min}$ is a parity family. Using Corollary 3 we therefore derive the following result.

Corollary 6 *Let $\mathcal{C} \subseteq 2^V$ be a lattice family and let f be a submodular function on $\mathcal{C}$. Then a set attaining the second smallest value of f over $\mathcal{C}$ can be obtained in oracle-polynomial time by solving $O(|V|^2)$ submodular function minimizations over lattice families.*

Grötschel, Lovász and Schrijver [7] reduce the problem of minimizing a submodular function over a triple family to a sequence of $O(|V|^3)$ submodular function minimizations over lattice families. By using the decomposition (1) or (2), one can in fact decrease the number of submodular function minimizations to $O(|V|^2)$. Their approach is in the spirit of Gomory and Hu's algorithm [5] to find a compact representation of minimum cuts between all pairs of vertices of a weighted undirected graph and the procedure of Padberg and Rao [12] to find a minimum-weight T-odd cut. At the heart of these methods is the following lemma which is a generalization of the notion of *uncrossing* introduced by Gomory and Hu.

Lemma 7 (The Uncrossing Lemma) *Let $\mathcal{C} \subseteq 2^V$ be a lattice family and $\mathcal{G} \subseteq \mathcal{C}$ be a triple family. Let f be a submodular function defined on $\mathcal{C}$. Let X minimize f over $\mathcal{G}$ and let A minimize f over $\mathcal{C}$. Then, if $A \notin \mathcal{G}$ then either $X \cup A$ or $X \cap A$ minimizes f over $\mathcal{G}$.*

In actual algorithmic applications, variations of this lemma are used, where the definition of A is somewhat different but the conclusion is identical.

Proof: By submodularity of f on $\mathcal{C}$, we have

$$f(A) + f(X) \geq f(A \cap X) + f(A \cup X). \tag{3}$$

Since A minimizes f over $\mathcal{C}$ we have $f(A \cup X) \geq f(A)$ and $f(A \cap X) \geq f(A)$. Combined with (3), this implies that $f(A \cap X) \leq f(X)$ and $f(A \cup X) \leq f(X)$. Therefore, we only need to prove that either $X \cap A$ or $X \cup A$ belongs to $\mathcal{G}$. This follows from the definition of the triple family $\mathcal{G}$ as applied to $X \in \mathcal{G}$ and $A \in \mathcal{C} \setminus \mathcal{G}$. □

If we replace the triple family by a parity family in the Uncrossing Lemma, it is no longer valid, as described in the following example. Let $V = \{1, 2\}, \mathcal{C} = 2^V$, $\mathcal{G} = \{\{2\}\}$ and $f(S) = \sum_{i \in S} a_i$ where $a_1 = -1$ and $a_2 = 1$. Since $\mathcal{C} \setminus \mathcal{G} = \{\emptyset, \{1\}, V\}$ is a chain of cardinality 3, $\mathcal{G}$ is a parity subfamily of the lattice family $\mathcal{C}$, but $\mathcal{G}$ is not a triple family. f is modular and, hence, submodular. By inspection, we get that $A = \{1\} \notin \mathcal{G}$ (uniquely) minimizes f over $\mathcal{C}$ and that $X = \{2\}$ (uniquely) minimizes f over $\mathcal{G}$. But neither $X \cup A = \{1, 2\}$ or $X \cap A = \emptyset$ minimize f over $\mathcal{G}$ (in fact, they are not even members of $\mathcal{G}$).

The example above clearly demonstrates that an algorithmic approach based on the Uncrossing Lemma will not work for all parity families. Our approach is based on the characterization described in Theorem 1, which is valid for any parity family. In the case where $\mathcal{C} \setminus \mathcal{G}$ is a chain, Theorem 1 is easily derived. Indeed, let us assume that $\mathcal{C} \setminus \mathcal{G} = \{C_1, \cdots, C_k\}$ where $C_1 \subset C_2 \subset \ldots \subset C_k$. For simplicity of notation, let $C_0 = \emptyset$ and $C_{k+1} = V$. Given a set $S^* \notin \{\emptyset, V\}$ minimizing f over $\mathcal{G}$, let $l = \max\{i : C_i \subseteq S^*\}$. Since $S^* \in \mathcal{G}$, we must have $S^* \setminus C_l \neq \emptyset$. Also, by definition of l, $C_{l+1} \setminus S^* \neq \emptyset$. Let $a \in S^* \setminus C_l$ and $b \in C_{l+1} \setminus S^*$. It is easy to see that $S^* \in \mathcal{C}_{ab}$ and $\mathcal{C}_{ab} \subseteq \mathcal{G}$. Thus S^* must minimize f over $\mathcal{C}_{ab}$. Observe that, in this special case, the submodularity of f is not exploited.

3.1 Symmetric Submodular Functions and Families

In this section, we consider the special case in which the submodular function is symmetric:

$$f(S) = f(V \setminus S) \quad \text{for all } S.$$

The best known example of a symmetric submodular function is given by the cut function of an undirected graph. More precisely, let $G = (V, E)$ be an undirected graph and let $c : E \to \mathcal{Z}^+$ be a weight function on the edges of G. For any set $S \subseteq V$, let $f(S) = \sum_{e \in \delta(S)} c_e$ where $\delta(S) = \{(i, j) \in E : i \in S, j \notin S\}$ is the cutset induced by S. In particular, $f(\emptyset) = f(V) = 0$. The cut function f is symmetric and is submodular on any lattice family $\mathcal{C} \subseteq 2^V$. Using the compact representation of a lattice family in terms of a digraph [8, Section 10.3], the minimum of the cut function f over any lattice family can be obtained by solving a single minimum s-t cut problem in a related digraph.

Most properties of the cut function also apply to any symmetric submodular function f. In particular, any symmetric submodular function admits a so-called *cut-equivalent tree* as was shown by Gomory and Hu [5] for the cut function. A cut equivalent tree is a tree $H = (V, T)$ on the groundset V such that a minimum set minimizing f over 2^V_{st} for any two elements s and t corresponds to either connected component obtained by removing an edge along the path from s to t in H. A cut equivalent tree can be obtained by solving $|V| - 1$ submodular function minimizations over 2^V_{st} [5].

Since f is symmetric, we can also restrict our attention to symmetric families of sets, i.e. families $\mathcal{C}$ such that $S \in \mathcal{C}$ iff $V \setminus S \in \mathcal{C}$. In the case of a symmetric *lattice* family $\mathcal{C}$, notice first that the empty set and V belong to $\mathcal{C}$: if $A \in \mathcal{C}$ then $V \setminus A$ is also in $\mathcal{C}$, and this implies that $A \cup (V \setminus A) = V$ and $A \cap (V \setminus A) = \emptyset$ belong to $\mathcal{C}$. Moreover, the minimum of a symmetric submodular function over a nonempty symmetric lattice family is trivially attained by the empty set, since $2f(A) = f(A) + f(V \setminus A) \geq f(V) + f(\emptyset) = 2f(\emptyset)$. Observe also that if $\mathcal{C}$ is a lattice family then its symmetrized family $\mathcal{C}_{sym} = \{S : S \in \mathcal{C}$ or $V \setminus S \in \mathcal{C}\}$ is not necessarily a lattice family (consider, for example, 2^V_{st}), but the following lemma implies that $\mathcal{C}_{sym}$ is a parity subfamily of 2^V.

Lemma 8 *Let $\mathcal{G}$ be a parity subfamily of a symmetric lattice family $\mathcal{C} \subseteq 2^V$. Then $\mathcal{G}_{sym} \subseteq \mathcal{C}$ is a symmetric parity family where $\mathcal{G}_{sym} = \{S \in \mathcal{C} : S \in \mathcal{G}$ or $V \setminus S \in \mathcal{G}\}$.*

Proof: It is easy to see that $\mathcal{G}_{sym}$ is symmetric. To show that it is a parity family, assume that $A, B \in \mathcal{C} \setminus \mathcal{G}_{sym}$. By definition, $A, B, V \setminus A$ and $V \setminus B$ belong to $\mathcal{C} \setminus \mathcal{G}$. Hence, $A \cup B \in \mathcal{G}$ iff $A \cap B \in \mathcal{G}$, and $V \setminus (A \cup B) \in \mathcal{G}$ iff $V \setminus (A \cap B) \in \mathcal{G}$. If $A \cup B \in \mathcal{C} \setminus \mathcal{G}_{sym}$ (the other case is identical), the above conditions imply that $A \cap B$ and $V \setminus (A \cap B)$ are both in $\mathcal{C} \setminus \mathcal{G}$, implying that $A \cap B \in \mathcal{C} \setminus \mathcal{G}_{sym}$. $\square$

Without loss of generality, then, when minimizing a symmetric submodular function over a parity family $\mathcal{G}$, we assume that this parity family is symmetric. To make the minimization problem non-trivial, we further assume that $\emptyset, V \notin \mathcal{G}$. The following lemma gives alternate characterizations of such families.

Lemma 9 *Let $\mathcal{C} \subseteq 2^V$ be a symmetric lattice family and let $\mathcal{G}$ be a symmetric subfamily of $\mathcal{C}$ with $\emptyset, V \notin \mathcal{G}$. Then the following are equivalent:*

(i). $\mathcal{G}$ is a parity family.

(ii). $\mathcal{G}$ satisfies: If $A, B \in \mathcal{C} \setminus \mathcal{G}$ and A, B are disjoint then $A \cup B \in \mathcal{C} \setminus \mathcal{G}$.

(iii). $\mathcal{G}$ satisfies: For A, B disjoint, if, among A, B and $A \cup B$, two sets belong to $\mathcal{C} \setminus \mathcal{G}$ then the third set also belongs to $\mathcal{C} \setminus \mathcal{G}$.

(iv). $\mathcal{G}$ is a triple family.

Proof: We show that $(iv) \Rightarrow (i) \Rightarrow (ii) \Rightarrow (iii) \Rightarrow (iv)$. It is clear that $(iv) \Rightarrow (i)$. Also, $(i) \Rightarrow (ii)$ since the definition of a parity family $\mathcal{G}$ applied to disjoint sets A and B in $\mathcal{C} \setminus \mathcal{G}$ implies that $A \cup B \in \mathcal{C} \setminus \mathcal{G}$ iff $\emptyset \in \mathcal{C} \setminus \mathcal{G}$.

We next show that $(ii) \Rightarrow (iii)$. Assume that $\mathcal{G}$ satisfies (ii) but not (iii). Thus there must exist disjoint sets A and B such that $A \in \mathcal{C} \setminus \mathcal{G}$, $A \cup B \in \mathcal{C} \setminus \mathcal{G}$ but $B \in \mathcal{G}$. Condition (ii) applied to A and $V \setminus (A \cup B)$ implies that $A \cup (V \setminus (A \cup B)) = V \setminus B \in \mathcal{C} \setminus \mathcal{G}$, a contradiction since $B \in \mathcal{G}$.

Finally, we show that $(iii) \Rightarrow (iv)$. Assume that $\mathcal{G}$ satisfies (iii) and consider two sets A and B in $\mathcal{C}$ such that exactly one out of $A, B, A \cup B$ and $A \cap B$ belongs to $\mathcal{G}$, say set X. Consider now condition (iii) applied to the triplets $\{A \setminus B, A \cap B, A\}$, $\{B \setminus A, A \cap B, B\}$, $\{A \setminus B, B, A \cup B\}$ and $\{B \setminus A, A, A \cup B\}$. The two triplets involving the set X imply that both $A \setminus B$ and $B \setminus A$ belong to $\mathcal{G}$, but the two other triplets imply that they belong to $\mathcal{C} \setminus \mathcal{G}$. This is a contradiction. $\square$

In the case of the cut function, families satisfying condition (ii) of Lemma 9 were defined as *proper* families by Goemans and Williamson [4]. Examples of proper families or proper cuts include:

- T-odd cuts. Given an even set T of vertices, a T-odd cut is a set $S \subseteq V$ such that $|S \cap T|$ is odd. The minimum T-odd cut problem arises as the separation problem over the matching or T-join polytopes. Padberg and Rao [12] have shown that minimum T-odd cuts can be obtained through a sequence of $O(|V|)$ minimum s-t cut problems. More precisely, they have shown that a minimum T-odd cut is induced by one of the edges of the Gomory-Hu cut tree. This immediately implies the existence of two vertices a and b as claimed in Theorem 1.
- More generally, families $\mathcal{G} = \{S : \sum_{i \in S} a_i \not\equiv 0 \pmod{p}\}$, where $a_i \in \mathbb{Z}$ and $p \in \mathbb{Z}_+$.
- Generalized Steiner cuts. Given sets $T_1, \ldots, T_k \subseteq V$ of terminals, a generalized Steiner cutset is induced by a set S such that $S \cap T_i \neq \emptyset$ and $T_i \setminus S \neq \emptyset$ for some S.

Gabow et al. [3, Section 5.1] have shown that any minimum proper cut problem can be solved by first constructing a cut equivalent tree. The result is described in the following theorem. It also applies to general symmetric submodular functions.

Theorem 10 (Gabow et al. [3]) *Let f be the cut function of a weighted undirected graph $G = (V, E)$ and let $\mathcal{G}$ be any proper family of cuts. Then the minimum of f over $\mathcal{G}$ can be obtained by taking the smallest cut in $\mathcal{G}$ among the cuts induced by a cut equivalent tree.*

This result does not follow immediately from Theorem 1 since there might be several minimum cuts between two vertices s and t, some of which are not in $\mathcal{G}$. Nevertheless, Theorem 10 can be indirectly derived from Theorem 1, as demonstrated below. We should however point out that the proof in [3] is simpler and more direct.

Proof: Consider a cut equivalent tree $H = (V, T)$ corresponding to the weighted graph $G = (V, E)$. We define a new weighted graph G' obtained from G by adding a new edge e with weight $\epsilon > 0$ for each edge $e \in T$ of the cut equivalent tree. Observe that the cuts induced by the edges of the cut equivalent tree will increase by ϵ while all other cuts will increase by at least 2ϵ. Thus, for G', all minimum s-t cuts are induced by edges of the cut equivalent tree. From Theorem 1, we know the existence of vertices a and b such that the minimum proper cut for G' is a minimum a-b cut. As a result, any minimum proper cut for G' must be induced by an edge of the cut equivalent tree. Letting ϵ be arbitrarily small, we derive the existence of a minimum proper cut for G induced by one of the edges of the cut tree. $\square$

In the rest of this section, we discuss minimum cut problems which are not proper but which can still be solved in polynomial-time using our characterization. We define a T-even cut as a set $S \subseteq V$ such that $S \neq \emptyset$, $S \neq V$ and $|S \cap T|$ is even. We emphasize that we exclude $S = \emptyset$ and $S = V$ in this definition.

3.1.1 s-t T-even and s-t T-odd cuts

An s-t T-odd cut (resp. s-t T-even cut) for some prespecified vertices s and t is a T-odd cut (resp. T-even cut) containing s but not t. Consider the lattice family $2^V_{st} = \{S : s \in S, t \notin S\}$ and its subfamily $\mathcal{G} = \{S \in 2^V_{st} : |S \cap T| \text{ odd}\}$. It is easy to see that $\mathcal{G}$ is a parity subfamily of 2^V_{st}, and even a triple family. The algorithm of Grötschel et al. [7, 8] thus solves the s-t T-odd cut problem as a sequence of $O(|V|^2)$ minimum s-t cut problems. This also follows from Corollary 3. In a similar vein, finding a minimum s-t T-even cut can be formulated and solved the same way.

3.1.2 T-even cuts

A polynomial time algorithm to solve the minimum T-even cut problem was devised by Barahona and Conforti [1]. They reduce the problem to a sequence of $O(|V|^5)$ minimum s-t cut problems. Using an argument similar to (1), one can decrease the number of minimum s-t cut problems to $O(|V|^4)$.

The minimum T-even cut problem cannot be formulated directly as the minimization over a parity family. Indeed, if one minimizes over the parity family $\mathcal{G} = \{S : |S \cap T| \text{ even}\}$ then one would obtain $S^* = \emptyset$ or $S^* = V$ as the optimum solution. The minimum T-even cut is thus equivalent to the problem of minimizing the cut function f over the family $\mathcal{G} \setminus \{\emptyset, V\}$. Let $\mathcal{G}_{st} = \{S \in \mathcal{G} : s \in S, t \notin S\}$. Observe that $\mathcal{G}_{st}$ is a parity subfamily of 2^V_{st} and that

$$\mathcal{G} \setminus \{\emptyset, V\} = \bigcup_{t \in V \setminus \{s\}} (\mathcal{G}_{st} \cup \mathcal{G}_{ts})$$

where s is any vertex (we could also use a decomposition similar to (2)). This shows that the minimum T-even cut problem can be reduced to $O(|V|)$ minimum s-t T-even cut problems, and thus can be solved in $O(|V|^3)$ s-t cut minimizations by using Corollary 3. The decomposition of $\mathcal{G}$ used above can be applied whenever we need to exclude $\emptyset$ and V from the minimization over a parity family.

4 Proofs

In this section, we prove Theorem 1. We first establish a key lemma from which our main result will follow easily. The proof technique we use was inspired by a result of Williamson et al. [13]. In what follows, we assume that $\mathcal{C} \subseteq 2^V$ is a lattice family, that $\mathcal{G} \subseteq \mathcal{C}$ is a parity family, that f is a submodular function on $\mathcal{C}$ and that S^* minimizes f over $\mathcal{G}$. We further assume that $S^* \notin \{\emptyset, V\}$.

Lemma 11 *There exists $a \in S^*$ such that $f(A) \geq f(S^*)$ for all $A \subseteq S^*$, $A \in \mathcal{C}$ and $a \in A$.*

Proof: The proof is by contradiction. Suppose not. Then, for all $a \in S^*$, there exists $T_a \subset S^*$ such that $a \in T_a$, $T_a \in \mathcal{C}$ and $f(T_a) < f(S^*)$. For each $a \in S^*$, we can choose T_a to be a maximal such set. By definition of S^*, $T_a \notin \mathcal{G}$ for all $a \in S^*$.

Claim 11.1 *$\bigcap_{a \in I} T_a \in \mathcal{C} \setminus \mathcal{G}$ for any $\emptyset \neq I \subseteq S^*$.*

> Since $\bigcap_{a \in I} T_a \in \mathcal{C}$, observe that the claim would follow from the definition of S^* and a proof that $f(\bigcap_{a \in I} T_a) < f(S^*)$ for any I. Given a nonempty $I \subseteq S^*$, choose a *maximal* set $I' \subseteq I$ such that $f(D) < f(S^*)$ where $D = \bigcap_{a \in I'} T_a$. If $I' \neq I$, choose any $b \in I \setminus I'$. Notice that D, T_b, $D \cup T_b$ and $D \cap T_b$ all belong to $\mathcal{C}$. By the submodularity of f on the lattice family $\mathcal{C}$,
>
> $$2f(S^*) > f(D) + f(T_b) \geq f(D \cup T_b) + f(D \cap T_b). \tag{4}$$
>
> Since T_b was chosen to be maximal, $D \cup T_b$ must satisfy $f(D \cup T_b) \geq f(S^*)$. Inequality (4) now implies that $f(D \cap T_b) < f(S^*)$. But this contradicts the maximality of I' implying that $I' = I$, and proving the claim. $\diamond$

Claim 11.2 *Let I and J be two disjoint subsets of S^*. Then*

$$\left(\left\{\bigcap_{a \in I} T_a\right\} \cap \left\{\bigcup_{b \in J} T_b\right\}\right) \in \mathcal{C} \setminus \mathcal{G}.$$

We prove the claim by double induction on (i,j) where $i = |I|$ and $j = |J|$, according to the following ordering of the pairs: $(i_1, j_1) < (i_2, j_2)$ if either $i_1+j_1 < i_2 + j_2$, or $i_1 + j_1 = i_2 + j_2$ and $j_1 < j_2$. For notational convenience, we set $T_I = \bigcap_{a\in I} T_a$, $T^J = \bigcup_{b\in J} T_b$ and $T_I^J = T_I \cap T_J$.

The base case corresponding to pairs (i,j) with $j \leq 1$ was proved in Claim 11.1. We consider two cases.

Case 1: $i = 0, j > 1$.

Let $c \in J$. For simplicity, we denote $J \setminus \{c\}$ by $J - c$. Since $(0, j-1) < (0, j)$, by induction, we have that $T^{J-c} \in \mathcal{C}\setminus\mathcal{G}$. From the definition of the parity family $\mathcal{G}$ applied to T^{J-c} and T_c, we derive that

$$T^{J-c} \cup T_c = T^J \in \mathcal{C} \setminus \mathcal{G} \Leftrightarrow T^{J-c} \cap T_c = T_c^{J-c} \in \mathcal{C} \setminus \mathcal{G}.$$

By induction, since $(1, j-1) < (0, j)$, $T_c^{J-c} \in \mathcal{C} \setminus \mathcal{G}$ and, therefore, $T^J \in \mathcal{C} \setminus \mathcal{G}$.

Case 2: $i > 0, j > 1$.

Let $c \in J$. Notice that

$$T_I^J = T_I \cap T^J = T_I \cap (T^{J-c} \cup T_c) = (T_I \cap T^{J-c}) \cup (T_I \cap T_c) = T_I^{J-c} \cup T_{I+c}, \quad (5)$$

where $I + c = I \cup \{c\}$. From Claim 11.1, we have that $T_{I+c} \in \mathcal{C}\setminus\mathcal{G}$. By induction, since $(i, j-1) < (i, j)$, we have that $T_I^{J-c} \in \mathcal{C} \setminus \mathcal{G}$. From the parity of $\mathcal{G}$ and using (5), we get that

$$T_I^J \in \mathcal{C} \setminus \mathcal{G} \Leftrightarrow T_I^{J-c} \cap T_{I+c} = T_{I+c}^{J-c} \in \mathcal{C} \setminus \mathcal{G}.$$

However, the latter set does not belong to $\mathcal{G}$ by induction, since $(i+1, j-1) < (i, j)$. $\diamond$

For $I = \emptyset$ and $J = S^*$, Claim 11.2 implies that $\cup_{b\in S^*} T_b = S^* \in \mathcal{C} \setminus \mathcal{G}$, contradicting the choice of S^*. This completes the proof of the lemma. $\square$

Remark 1 *When $\mathcal{G}$ is defined to be $\{S \in \mathcal{C} : |S| \not\equiv q \pmod p\}$, where $q, p \in \mathbb{Z}^+$, we can give a shorter proof of $\cup_{a\in S^*} T_a \in \mathcal{C} \setminus \mathcal{G}$ using the inclusion-exclusion principle:*

Let $l = |S^*|$ and arbitrarily label the elements in S^* from 1 through l. By the inclusion-exclusion principle,

$$|\cup_{i=1}^l T_i| = \sum_{i=1}^{l} |T_i| - \sum_{i<j} |T_i \cap T_j| + \sum_{i<j<k} |T_i \cap T_j \cap T_k| - \cdots$$

By Claim 11.1, every term of the form $|T_i \cap \cdots \cap T_k|$ is congruent to $q \pmod p$, so we have

$$\begin{aligned} |\cup_{i=1}^{l} T_i| &\equiv \binom{l}{1} q - \binom{l}{2} q + \binom{l}{3} q - \cdots + (-1)^{l+1} \binom{l}{l} q \pmod p \\ &\equiv (1 - (1-1)^l)\, q \pmod p \\ &\equiv q \pmod p. \end{aligned}$$

$\diamond$

We can obtain a similar result as in Lemma 11 for $V - S^*$. Consider the submodular function $f'(S) = f(V - S)$ on the lattice family $\mathcal{C}' = \{S : V - S \in \mathcal{C}\}$. Notice that $V - S^*$ minimizes f' over the parity family $\mathcal{G}' = \{S : V - S \in \mathcal{G}\}$. From Lemma 11, we then derive:

Lemma 12 *There exists $b \in V - S^*$ such that $f(B) \geq f(S^*)$ for all $B \supseteq S^*$, $B \in \mathcal{C}$ and $b \notin B$.*

We are now ready to prove Theorem 1.

Proof of Theorem 1:

Let a be the element of S^* referred to in Lemma 11 and b the element of $V - S^*$ referred to in Lemma 12. Let W be a set minimizing f over $\mathcal{C}_{ab}$. Thus $f(S^*) \geq f(W)$ since $S^* \in \mathcal{C}_{ab}$. By the submodularity of f,

$$2f(S^*) \geq f(S^*) + f(W) \geq f(S^* \cap W) + f(S^* \cup W). \tag{6}$$

By the choice of a, $f(S^* \cap W) \geq f(S^*)$ and, by the choice of b, $f(S^* \cup W) \geq f(S^*)$. Therefore, all inequalities of (6) must be equalities, implying that $f(S^*) = f(W)$. This implies that S^* also minimizes f over $\mathcal{C}_{ab}$. $\square$

By strengthening Lemma 11, one can also derive Theorem 2.

Lemma 13 *Let S^* be a minimal set minimizing f over $\mathcal{G}$. Then there exists $a \in S^*$ such that $f(A) > f(S^*)$ for all $A \subset S^*$, $A \in \mathcal{C}$ and $a \in A$.*

Proof: The proof is almost identical to the proof of Lemma 11. Suppose the claim is not true. Then, for all $a \in S^*$, there exists $T_a \subset S^*$ such that $a \in T_a$, $T_a \in \mathcal{C}$ and $f(T_a) \leq f(S^*)$. For each $a \in S^*$, we can choose T_a to be a maximal such set. By definition of S^*, $T_a \in \mathcal{C} \setminus \mathcal{G}$ for all $a \in S^*$. Below we present a modification of the proof of Claim 11.1. Claim 11.2 and its proof remain unaltered.

Claim 13.1 *$\bigcap_{a \in I} T_a \in \mathcal{C} \setminus \mathcal{G}$ for any $\emptyset \neq I \subseteq S^*$.*

Since $\bigcap_{a \in I} T_a \in \mathcal{C}$, observe that the claim would follow from the definition of S^* and a proof that $f(\bigcap_{a \in I} T_a) \leq f(S^*)$ for any I. Given a nonempty $I \subseteq S^*$, choose a *maximal* set $I' \subseteq I$ such that $f(D) \leq f(S^*)$ where $D = \bigcap_{a \in I'} T_a$. If $I' \neq I$, choose any $b \in I \setminus I'$. Notice that D, T_b, $D \cup T_b$ and $D \cap T_b$ all belong to $\mathcal{C}$. By the submodularity of f on the lattice family $\mathcal{C}$,

$$2f(S^*) \geq f(D) + f(T_b) \geq f(D \cup T_b) + f(D \cap T_b). \tag{7}$$

We claim that $f(D \cup T_b) \geq f(S^*)$. Indeed, either $D \cup T_b = S^*$ in which case $f(D \cup T_b) = f(S^*)$ or $D \cup T_b \subset S^*$ in which case the maximality of T_b implies that $f(D \cup T_b) > f(S^*)$. Inequality (7) now implies that $f(D \cap T_b) \leq f(S^*)$, contradicting the maximality of I'. $\diamond$

For $I = \emptyset$ and $J = S^*$, Claim 11.2 implies that $\cup_{b \in S^*} T_b = S^* \in \mathcal{C} \setminus \mathcal{G}$, contradicting the choice of S^*. This completes the proof of the lemma. $\square$

Proof of Theorem 2:

Let S^* be a minimal set minimizing f over $\mathcal{G}$. Let a be the element of S^* referred to in Lemma 13 and b the element of $V - S^*$ referred to in Lemma 12. Let W be a set minimizing f over $\mathcal{C}_{ab}$. As in the proof of Theorem 1, we derive that all inequalities in (6) are satisfied at equality, i.e. $f(S^*) = f(W) = f(S^* \cap W) = f(S^* \cup W)$, implying that S^* minimizes f over $\mathcal{C}_{ab}$. Moreover, $S^* \cap W = S^*$ for otherwise $S^* \cap W \in \mathcal{C}_{ab}$ and $f(S^* \cap W) = f(S^*)$ will contradict the choice of a. Thus, every set W minimizing f over $\mathcal{C}_{ab}$ contains S^* and hence S^* is the unique minimal set minimizing f over $\mathcal{G}$. $\square$

5 Algorithmic Implications

In this section, we deduce the algorithmic consequences of Theorems 1 and 2. In what follows, we assume that the lattice family $\mathcal{C}$ is compactly encoded by a digraph as described in [8, Section 10.3], that the parity family is given by a membership oracle and that the submodular function is given by an oracle returning $f(S)$ for each query $S \subseteq V$. Let β be an upper bound on the size of $f(S)$ for any set S.

As mentioned previously, the minimization of a submodular function f over a lattice family $\mathcal{C} \subseteq 2^V$ can be done in oracle-polynomial time [6]. We claim further that the algorithm can be used to find a *minimal* set minimizing f over the lattice family. Indeed, letting $g(S) = f(S) + \epsilon|S|$, we observe that g is submodular and that, for ϵ sufficiently small, every set minimizing g will be a minimal set minimizing f. The scalar ϵ, for example, can be set to be $\frac{1}{|V|}2^{-2\beta}$ since the definition of β implies that $|f(S_1) - f(S_2)| \geq 2^{-2\beta}$ whenever $S_1 \neq S_2$. Observe that the size of $g(S)$ is $O(\beta + \log|V|)$, implying that the algorithm of

Grötschel et al. [6] as applied to g will find a minimal set minimizing f in oracle-polynomial time.

Corollary 3 follows easily from Theorem 2. The algorithm proceeds as follows. We first find a minimal set S_{ab} minimizing f over the lattice family $\mathcal{C}_{ab}$ for every a and b in V. The compact representation for $\mathcal{C}_{ab}$ can easily be obtained in linear time from the compact representation for $\mathcal{C}$. We then select in the collection $\{\emptyset, V\} \cup \{S_{ab} : a, b \in V\}$ a set $S \in \mathcal{G}$ having the smallest value $f(S)$. Theorem 2 guarantees that this set S is a minimal set minimizing f over the family $\mathcal{G}$. This algorithm requires the computation of $O(|V|^2)$ submodular function minimizations over lattice families.

We would like to make a few remarks. First, the algorithm can also be used to output all minimal sets minimizing f over $\mathcal{G}$ in the same amount of time. In particular, observe that Theorem 2 implies that there are at most $|V|(|V|-1)$ minimal sets minimizing f over a parity family. Also, we would like to emphasize that the collection $\{\emptyset, V\} \cup \{S_{ab} : a, b \in V\}$ is independent of which parity subfamily of $\mathcal{C}$ is being considered. This implies that we can find a minimal set minimizing f over *any* parity subfamily of $\mathcal{C}$ easily, once this collection is constructed.

6 Approximately Minimum Cuts

The techniques developed in this paper can also be used in other settings. Consider the recent result of Nagamochi, Nishimura and Ibaraki [11] which says that there are at most $\binom{n}{2}$ cuts in an undirected graph whose value is less than 4/3 times the minimum cut value. This result follows from the following characterization of such cuts.

Theorem 14 *Let f be the cut function of a weighted undirected graph $G(V, E)$ and let $\mathcal{H} = \{S \subseteq V \setminus \{1\} : f(S) < \frac{4}{3} \min_{\emptyset \neq T \subset V} f(T)$. Then for any $S \in \mathcal{H}$, there exist $a, b \in V \setminus \{1\}$ (a and b are not necessarily distinct) such that S is the unique minimal set in $\mathcal{H}$ containing both a and b.*

There are at most $\binom{n-1}{2} + n - 1 = \binom{n}{2}$ different choices for a, b, implying the result of Nagamochi et al. that $|\mathcal{H}| \leq \binom{n}{2}$. The proof of Theorem 14 is much in the spirit of the proof of Theorem 1. The non-existence of the vertices a, b for a given set S can be seen to imply the existence of vertices $x, y, z \in S$ and sets $X, Y, Z \in \mathcal{H}$ such that $x \in (Y \cap Z \setminus X)$, $y \in (Z \cap X \setminus Y)$ and $z \in (X \cap Y \setminus Z)$. However, the triple sbmodular inequality [10, Ex. 6.48] then implies that

$$\begin{aligned} 4 \min_{\emptyset \neq T \subset V} f(T) &\leq f(Y \cap Z \setminus X) + f(Z \cap X \setminus Y) + f(X \cap Y \setminus Z) + f(V \setminus (X \cup Y \cup Z)) \\ &\leq f(X) + f(Y) + f(Z) \\ &< 4 \min_{\emptyset \neq T \subset V} f(T), \end{aligned}$$

which is a contradiction.

Acknowledgements

The first author would like to thank Michele Conforti for fruitful discussions, András Frank for bringing to his attention the applicability of the algorithm of Grötschel et al. to the minimum s-t T-odd cut problem, and András Benczur for discussions which lead to the alternate proof of the result of Nagamochi et al.

References

[1] F. Barahona and M. Conforti, "A construction for binary matroids", *Discrete Mathematics*, **66**, 213–218, 1987.

[2] S. Fujishige, "Submodular functions and optimization", Annals of Discrete Mathematics 47, North-Holland, Amsterdam, 1991.

[3] H.N. Gabow, M.X. Goemans and D.P. Williamson, "An efficient approximation algorithm for the survivable network design problem", *Proc. of the Third MPS Integer Programming and Combinatorial Optimization Conference*, Erice, Italy, 57–74, 1993.

[4] M.X. Goemans and D.P. Williamson, "A general approximation technique for constrained forest problems", *SIAM Journal on Computing*, to appear. A preliminary version has appeared in the *Proc. of the Third Annual ACM-SIAM Symposium on Discrete Algorithms*, Orlando, Florida, 307–316, 1992.

[5] R.E. Gomory and T.C. Hu (1961), "Multi-terminal network flows", *Journal of the Society for Industrial and Applied Mathematics*, **9**, 551–570, 1961.

[6] M. Grötschel, L. Lovász and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization", *Combinatorica*, **1**, 169–197, 1981.

[7] M. Grötschel, L. Lovász and A. Schrijver, "Corrigendum to our paper "The ellipsoid method and its consequences in combinatorial optimization"", *Combinatorica*, **4**, 291–295, 1984.

[8] M. Grötschel, L. Lovász and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1988.

[9] L. Lovász, "Submodular functions and convexity", in: A. Bachem, M. Grötschel and B. Korte (eds.), *Mathematical Programming: The State of the Art, Bonn, 1982*, Springer, Berlin, 235–257, 1983.

[10] L. Lovász, "Combinatorial problems and exercises", Akadémiai Kiadó, 1993.

[11] H. Nagamochi, K. Nishimura and T. Ibaraki, "Computing all small cuts in undirected networks", *Proceedings of the 5th ISAAC, Beijing, P.R. China, 1994*, Lecture Notes in Computer Science, **834**, D.-Z. Du and X.-S. Zhang, eds., Springer-Verlag, 190–198 (1994).

[12] M.W. Padberg and M.R. Rao, "Odd minimum cut-sets and b-matchings", *Mathematics of Operations Research*, **7**, 67–80, 1982.

[13] D. Williamson, M.X. Goemans, M. Mihail and V. Vazirani, "A Primal-Dual Approximation Algorithm for Generalized Steiner Network Problems", *Proc. of the 25th ACM Symposium on the Theory of Computing*, San Diego, CA, 708–717, 1993.