Submodular problems - approximations and algorithms

Dorit S. Hochbaum *
email: hochbaum@ieor.berkeley.edu

University of California, Berkeley

Abstract. We show that any submodular minimization (SM) problem defined on linear constraint set with constraints having up to two variables per inequality, are 2-approximable in polynomial time. If the constraints are monotone (the two variables appear with opposite sign coefficients) then the problems of submodular minimization or supermodular maximization are polynomial time solvable. The key idea is to link these problems to a submodular s,t-cut problem defined here. This framework includes the problems: SM-vertex cover; SM-2SAT; SM-min satisfiability; SM-edge deletion for clique, SM-node deletion for biclique and others. We also introduce here the submodular closure problem and show that it is solvable in polynomial time and equivalent to the submodular cut problem. All the results are extendible to multi-set where each element of a set may appear with a multiplicity greater than 1. For all these NP-hard problems 2-approximations are the best possible in the sense that a better approximation factor cannot be achieved in polynomial time unless NP=P. The mechanism creates a relaxed “monotone” problem, solved as a submodular closure problem, the solution to which is mapped to a half integral super-optimal solution to the original problem. That half-integral solution has the persistency property meaning that integer valued variables retain their value in an optimal solution. This permits to delete the integer valued variables, and restrict the search of an optimal solution to the smaller set of remaining variables.

1 Introduction

Let \( V \) be a finite nonempty set of cardinality \( n \). A nonnegative function \( f \) defined on the subsets of \( V \) is said to be submodular if it satisfies for all \( X,Y \subseteq V \),

\[
 f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y).
\]

We consider here submodular minimization on linear constraints. For any binary vector \( \mathbf{x} = \{x_i\}_{i=1}^n \) the corresponding set \( X \) is the characteristic set of \( \mathbf{x} \). Namely, \( X = \{i|x_i = 1\} \). The problem of submodular minimization (SM) on \( m \) linear constraints, for an \( m \times n \) matrix \( A \) and a vector \( \mathbf{b} \) is,

* Research supported in part by NSF award No. DMI-0620677 and CBET-0736232.
A set $X$ is said to be feasible, if the corresponding binary vector $x$ satisfies $Ax \geq b$. Submodular function $f$ is said to be monotone if $f(S) \leq f(T)$ for any $S \subseteq T$, and normalized if $f(\emptyset) = 0$. In all problems studied here the submodular functions are monotone and normalized.

Submodular minimization on multi-sets allows the multi-sets to contain elements with multiplicity larger than 1. So for an integer, nonnegative, vector $x$, the corresponding multiset is the collection of pairs $X = \{(i, q_i)|x_i = q_i\}$ meaning that the set $X$ contains element $i$ $q_i$ times. All properties of submodular functions extend easily to multi-sets, with the definition of containment, $X_1 \subseteq X_2$ indicating that for all $(i, q_i) \in X_1$, $(i, q'_i) \in X_2$ with $q_i \leq q'_i$.

The submodular multi-set minimization problem is then,

$$\min \ f(X)$$

subject to $Ax \geq b$

$x_j$ binary $j \in V$.

Submodular optimization problems are only harder to optimize, or approximate, than their linear optimization counter-parts since linear functions are also submodular. Nevertheless, it is demonstrated here that all polynomial time 2-approximation algorithms for NP-hard linear optimization in integers over constraints with up to two variables per inequality, extend to 2-approximations for the respective submodular problems, also running in polynomial time. The complexity of the linear problem is the running time for a minimum cut in a respective graph, whereas for the submodular problem this is substituted by the run time of the submodular cut problem, SM-cut, introduced here.

A solution corresponding to a cut is associated with a partition of $V$ to a source set and sink set. The source sets corresponding to cuts form a ring since their union and intersection are also source sets of cuts. Submodular minimization over a ring family, or over all subsets, was shown first to be solved in strongly polynomial time by Grötschel, Lovász, and Schrijver in [GLS88]. Combinatorial, strongly polynomial algorithms were given later by Schrijver, [Sch00] and by Iwata, Fleischer, and Fujishige [IFF01]. The current fastest strongly polynomial algorithm on a ring family was given by Orlin [Or09], and later by Iwata and Orlin, [IO09]. The class of submodular minimization problems presented here is therefore of equivalent complexity to that of the respective class of linear minimization problems in that the routine of minimum $s,t$-cut on an associated graph employed in the linear case, is replaced by the SM-cut on the same constructed graph for the respective SM problem.

There is a large body of work on approximations in general, and 2- approximations in particular, which is not reviewed here. Instead we only address a couple of papers within the approximation literature. The technique we use here
is an extension of that of Hochbaum et al. [HMNT93] for linear integer minimization on two variables per inequality. Using the local-ratio technique Bar-Yehuda and Rawitz [BR01] offered an alternative approach for 2-approximating such problems. This has the advantage when $u_j$s are very large. The construction of the graph here and in [HMNT93] affects the number of arcs in the constructed graph in a quadratic factor of $u_j$ whereas the local-ratio algorithm is affected by only a linear factor. The algorithm we use for solving the SM-cut problem, of [IN09], is not affected by the number of arcs, only by the number of nodes. Hence this complexity improvement does not extend to the SM case. That approach also does not lend itself to an extension to the submodular minimization case in that it does not have the persistence property that the half integral superoptimal solution obtained here and in [HMNT93] has. Koufogiannakis and Young [KY09] devised approximations for SM- “covering” problems based on the frequency technique (called maximal dual feasible technique in [Hoc97] Ch. 3). This applies to submodular minimization on “covering-type” matrices with nonnegative entries with “monotone” property. That “monotone” property is unrelated to the monotone constraints addressed here.

The technique described here applies to a large class of NP-hard submodular optimization problems including: The submodular minimizations of vertex cover; minimum 2-SAT; minimum node deletion biclique; minimum edge deletion clique; min SAT; and any submodular optimization on constraints each including up to two variables.

Two new polynomial time submodular optimization problems are introduced here. One is the submodular $s, t$-cut problem and the other is the closely related (and shown equivalent) submodular closure problem which is to maximize (or to minimize) the sum of supermodular revenues minus submodular costs (or minimize submodular costs minus supermodular revenues). The algorithm for solving the submodular cut problem is the key subroutine in solving all the other problems presented here.

One indication that the submodular case is possibly harder to approximate than the linear case has been established for the vertex cover problem: Hochbaum conjectured in [Hoc83] that the lower bound on approximability of the vertex cover problem is $2 - \epsilon$. The tightest lower bound established to date on the approximability of vertex cover is 1.36 (Dinur and Safra [DS02]), whereas for the submodular monotone analog it is $2 - \epsilon$ (Goel et al. [GKTW09])). The latter closes the gap between the lower bound and the factor 2 approximation, as conjectured in [Hoc83], for submodular vertex cover. The results presented here mean that for all NP-hard submodular minimization problems on two variables per inequality, there is a polynomial time 2-approximation algorithm, which cannot be improved. This is because the vertex cover problem is as general as the entire class of these problems (as shown in Section 5.3).
2 Notations and preliminaries

Given an integer matrix \( A_{m \times n} \) so that each row contains at most two non-zeroes. If the two non-zeroes are of opposite signs then we call the matrix monotone. We define the class of submodular minimization problems on constraints with at most two variables per constraint, SM2:

\[
\text{(SM2)} \quad \min \quad f(X) \\
\text{subject to} \quad Ax \geq b \\
0 \leq x_j \leq u_j \quad \text{integer} \quad j \in V.
\]

Problem SM2 is said to be monotone if the corresponding matrix \( A \) is monotone. If all \( u_j = 1 \) we call the problem a binary SM2.

For a directed graph \( G = (V, A) \) and \( B, D \subseteq V \), we denote by \( (B, D) \) the set of arcs from nodes in \( B \) to nodes in \( D \), \( (B, D) = \{(i, j) \mid i \in B, j \in D\} \). Note that \( B \) and \( D \) need not be disjoint and can be equal. In \( G = (V, A) \), a set of nodes \( D \subseteq V \) is said to be closed if all the successors of the nodes in \( D \) are also in \( D \). In other words, the transitive closure of \( D \), forming all the nodes reachable from nodes of \( D \) along a directed path in \( G \), is equal to \( D \).

An arc-capacitated graph \( G_{st} = (V \cup \{s, t\}, A \cup A_s \cup A_t) \) with \( A_s \) the set of arcs adjacent to source \( s \) and \( A_t \) a set of arcs adjacent to sink \( t \), is said to be a closure graph if all arcs in \( A \) have infinite capacity.

A function \( f \) is said to be supermodular if for all \( X, Y \subseteq V \),

\[
f(X) + f(Y) \leq f(X \cap Y) + f(X \cup Y).
\]

3 Examples of submodular minimization problems solved here

The submodular vertex cover problem has been shown recently to have a 2-approximation, [IN09]. Here we describe a 2-approximation algorithm which applies to the entire family of submodular minimization (and supermodular maximization) with constraints with up to two variables each. The general purpose method used is an extension of the algorithm used by Hochbaum [Hoc83] for (linear) vertex cover, and later generalized by Hochbaum et al. [HMNT93] to all integer linear minimization on two variables per inequality, and later still by [Hoc02] to a restricted class of integer linear minimization on constraints with up to three variables per inequality.

One reason why the generalization works, is that the technique used, of monotonizing and binarizing, is a dual method manipulates constraints and is independent of the objective function. This also explains why the maximum frequency approximation to set cover can be extended to submodular set cover with the same approximation guarantee [KY09,IN09], whereas the greedy approximation algorithm, which is a primal approach, cannot be extended to the submodular version of the set cover, [IN09].
**Vertex cover.** The vertex cover problem is to find a subset of nodes in a graph $G = (V, E)$ so that each edge in $E$ has at least one endpoint in the subset.

\[
\begin{align*}
\min & \quad f(X) \\
\text{(SM-vertex-cover)} & \quad \text{subject to } x_i + x_j \geq 1 \text{ for all } [i, j] \in E \\
& \quad x_i \text{ binary } i \in V.
\end{align*}
\]

**Complement of maximum clique.** The maximum clique problem is a well-known optimization problem that is notoriously hard to approximate as shown by Håstad, [Ha96]. The problem is to find in a graph the largest set of nodes that forms a clique – a complete graph.

An equivalent statement of the clique problem is to find the complete subgraph which maximizes the number (or more generally, sum of weights) of the edges in the subgraph. When the weight of each edge is 1, then there is a clique of size $k$ if and only if there is a clique on $\binom{k}{2}$ edges. The inapproximability result for the node version extends trivially to this edge version as well.

The complement of this edge variant of the maximum clique problem is to find a minimum weight of edges to delete so the remaining subgraph induced on the non-isolated nodes is a clique. We define here the SM-edge deletion for clique. For a graph $G = (V, E)$, the submodular function $f(Z)$ is defined on the set of variables $z_{ij}$ for all edges $[i, j] \in E$. Let $x_j$ be a variable that is 1 if node $j$ is in the clique, and 0 otherwise. Let $z_{ij}$ be 1 if edge $[i, j] \in E$ is deleted.

\[
\begin{align*}
\min & \quad f(Z) \\
\text{(SM-Clique-edge-delete)} & \quad \text{subject to } 1 - x_i \leq z_{ij} \quad [i, j] \in E \\
& \quad 1 - x_j \leq z_{ij} \quad [i, j] \in E \\
& \quad x_i + x_j \leq 1 \quad [i, j] \notin E \\
& \quad x_j \text{ binary } j \in V \\
& \quad z_{ij} \text{ binary } [i, j] \in E.
\end{align*}
\]

This formulation has two variables per inequality and therefore 2-approximation follows immediately. The gadget and network for solving the monotonized SM-Clique-edge-delete problem are given in detail in [Hoc02].

**Node deletion biclique.** Here we consider the submodular minimization of node deletion in a bipartite graph so that the remaining nodes form a biclique (a complete bipartite graph.) This problem is in fact polynomial time solvable since, with the “monotonizing” transformation, an equivalent set of constraints is monotone. In the formulation given below $x_i$ assumes the value 1 if node $i$ is deleted from the bipartite graph, and 0 otherwise.

\[
\begin{align*}
\min & \quad f(X) \\
\text{(SM-Biclique-node-delete)} & \quad \text{subject to } x_i + x_j \geq 1 \text{ for edge } \{i, j\} \notin E \quad i \in V_1, j \in V_2 \\
& \quad x_j \in \{0, 1\} \text{ for all } j \in V.
\end{align*}
\]

Notice that this formulation is identical to that of the submodular vertex cover on a bipartite graph, and both are polynomial time solvable. Note also that the SM-node deletion clique is the same as SM-vertex cover and thus NP-hard, and has a polynomial time 2-approximation.
Minimum satisfiability. In the problem of minimum satisfiability, MIN-SAT, we are given a CNF satisfiability formula. The aim is to find an assignment satisfying the smallest number of clauses, or the smallest weight collection of clauses. The MINSAT problem was introduced by Kohli et. al. [KKM94] and was further studied by Marathe and Ravi [MR96]. The problem is NP-hard.

To see that the submodular minimum satisfiability SM-MINSAT problem can be formulated as SM2, and thus 2-approximable, we choose a binary variable \( y_j \) for each clause \( C_j \) and \( x_i \) for each literal. Let \( S^+(j) \) be the set of variables that appear unnegated and \( S^-(j) \) those that are negated in clause \( C_j \). The following formulation of MINSAT has two variables per inequality and is thus a special case of SM2:

\[
\begin{align*}
\text{(SM-MINSAT)} \quad \min & \quad f(Y) \\
\text{subject to} & \quad y_j \geq x_i \quad \text{for } i \in S^+(j) \quad \text{for clause } C_j \\
& \quad y_j \geq 1 - x_i \quad \text{for } i \in S^-(j) \quad \text{for clause } C_j \\
& \quad x_i, y_j \text{ binary for all } i, j.
\end{align*}
\]

It is interesting to note that the formulation is monotone when for all clauses \( C_j \), \( S^+(j) = \emptyset \) or in all clauses \( S^-(j) = \emptyset \). (In the latter case need to transform the \( x \) variables to \( x' \) with \( x' = -x \).) Indeed in these instances the boolean expression is uniform and the problem is trivially solved setting all variables to FALSE in the first case, or to TRUE in the latter case.

MIN-2SAT. The MIN-2SAT problem is defined for a 2SAT CNF with each clause containing at most two variables. The goal is to find a least weight collection of variables that are set to true, so that the respective 2SAT CNF is satisfied. Although finding a satisfying assignment to a 2SAT can be done in polynomial time, Even et al. [EIS76], finding an assignment that minimizes the number, or the weight, of the true variables in NP-hard.

Let \( X \) be the set of true variables, and \( x_i = 1 \) if the \( i \)th variable is set to true, and 0 otherwise.

\[
\begin{align*}
\text{(SM-MIN-2SAT)} \quad \min & \quad f(X) \\
\text{subject to} & \quad x_i + x_j \geq 1 \quad \text{for clause } (x_i \lor x_j) \\
& \quad x_i - x_j \geq 0 \quad \text{for clause } (x_i \lor \bar{x}_j) \\
& \quad x_i + x_j \leq 1 \quad \text{for clause } (\bar{x}_i \lor \bar{x}_j) \\
& \quad x_i \text{ binary for all } i = 1, \ldots, n.
\end{align*}
\]

Each constraint here has up to two variables and thus this problem is in the class SM2. Consequently we get for this problem a 2-approximation in polynomial time. The construction of the graph is given in [Hoc02].

Additional problems related to finding maximum biclique – a clique in a bipartite graph – are also formulated in two variables per inequality in [Hoc98]. Therefore all the corresponding submodular minimization problems are also either solved in polynomial time, if monotone, or have a polynomial time 2-approximation.
4 The submodular closure problem

The submodular closure problem is defined on a directed graph $G = (V, A)$, a partition of the set of nodes $V = V^+ \cup V^-$, a supermodular function $f_1()$ defined on subsets of $V^+$ and a submodular function $f^-()$ defined on subsets of $V^-$. The submodular closure problem is,

$$\text{(SM-closure)} \max_{D \subseteq V, D \text{ closed}} f_1(D \cap V^+) - f^-(D \cap V^-).$$

In minimization form the SM-closure problem is $\min_{D \subseteq V, D \text{ closed}} f^-(D \cap V^-) - f_1(D \cap V^+)$ Notice that the SM-closure problem involves both a supermodular and a submodular functions in the objective. The problem of submodular optimal closure is a generalization of the (linear) closure problem defined on a directed graph $G = (V, A)$. The closure requirement is represented as a set of monotone constraints:

$$\text{(max-closure)} \max \sum_{j \in V} w_j x_j$$

subject to $x_i - x_j \geq 0 \quad \forall (i, j) \in A,$

$x_j$ binary $j \in V.$

The linear max-closure problem induces a partition on $V$ with $V^+ = \{v \in V | w_v > 0\}$ and $V^- = \{v \in V | w_v \leq 0\}$. A non-binary integer version of the minimum closure problem with a convex separable objective replacing the linear objective was shown to be solvable in polynomial time by a parametric cut algorithm in Hochbaum and Queyranne [HQ03]. The SM-closure problem is shown next to be equivalent to a submodular cut problem, SM-cut.

Fig. 1. The submodular closure problem on a bipartite closure graph.
For a graph $G = (V, A)$ and a partition of $V$ to $V^+ \cup V^-$, we construct a corresponding $s,t$ graph, $G_{st}$, by adding source and sink nodes, $s$ and $t$, and connecting $s$ to nodes of $V^+ \subset V$ with a set of arcs $A_s = \{(s,i) | i \in V^+\}$, and connecting $t$ to nodes if $V^- = V \setminus V^+$ with a set of arcs $A_t = \{(j,t) | j \in V^-\}$.

$G_{st} = (V \cup \{s,t\}, A \cup A_s \cup A_t)$. A cut is a partition of the set of nodes $V$ to $S$ and $\bar{S} = V \setminus S$ with $\{s\} \cup S$ called the source set of the cut, and $\{t\} \cup \bar{S}$ called the sink set of the cut. A cut is said to be finite if $(S, \bar{S}) \neq \emptyset$, or equivalently, $S$ is closed in $G$. Given two submodular functions $f^+$ and $f^-$ defined on $V^+$ and $V^-$ respectively, we define the submodular cut problem, SM-cut, on the graph $G_{st}$ as follows,

$$\text{(SM-cut)} \quad \min_{D \subseteq V, D \text{ closed}} f^+(\bar{D} \cap V^+) + f^-(D \cap V^-).$$

This definition is analogous to that of the minimum $s,t$-cut problem on a closure graph. Note that for a finite cut $(\{s\} \cup S, \bar{S} \cup \{t\})$, the only arcs that participate in the cut are arcs $(\{s\}, \bar{S} \cap V^+)$ and $(S \cap V^-, \{t\})$. The connection between the submodular closure and SM-cut problems is established next:

**Theorem 1** The optimal solution to the SM-closure problem is also optimal for a respective SM-cut problem.

**Proof:** Since the function $f_1()$ is supermodular on subsets of $V^+$ then for any constant $C$ the function $f'(B) = C - f_1(B \cap V^+)$ is submodular on subsets of $V^+$. (We’ll choose $C$ large enough so the resulting function assumes nonnegative values.) To see that let $B_1, B_2 \subseteq V^+$,

$$f'(B_1) + f'(B_2) = 2C - [f_1(B_1 \cap V^+) + f_1(B_2 \cap V^+)].$$

Since $f_1()$ is supermodular,

$$f_1(B_1 \cap V^+) + f_1(B_2 \cap V^+) \leq f_1((B_1 \cup B_2) \cap V^+) + f_1((B_1 \cap B_2) \cap V^+) = C - f'(B_1 \cap B_2) + C - f'(B_1 \cap B_2).$$

Thus, $f'(B_1) + f'(B_2) \geq f'(B_1 \cup B_2) + f'(B_1 \cap B_2)$, and $f'(\cdot)$ is submodular as claimed. Now,

$$\max_{D \subseteq V, D \text{ closed}} f_1(D \cap V^+) - f^-(D \cap V^-) = \max_{D \subseteq V, D \text{ closed}} C - f'(\bar{D} \cap V^+) - f^-(D \cap V^-) = C - \min_{D \subseteq V, D \text{ closed}} f'(\bar{D} \cap V^+) + f^-(D \cap V^-).$$

The latter minimization problem is the SM-cut problem. For the constant $C = f_1(V^+)$ we denote the submodular function $f'(\cdot)$ by $f^+(\cdot)$. We thus showed that an optimal solution to the SM-closure problem is also optimal for the SM-cut problem.

In Figure 1 we illustrate the SM-cut problem for a bipartite instance of $G$, where each arc adjacent to source or sink is shown with the respective singleton node function value.
As noted in the introduction, the feasible sets for SM-cut form a ring as the intersection and union of any source sets of two closed sets is also a closed set. As such the SM-cut is solved in polynomial time as submodular minimization on a ring.

5 Solving submodular integer programs SM2

Here we show that any monotone SM2 problem is equivalent to a SM-closure problem on a graph on number of nodes which is \( O(\sum_{i \in V} u_i) \). Note that if the range of the variables is not a polynomial quantity then the size of the graph is pseudopolynomial. This run time however cannot be made polynomial, unless \( \text{NP} = \text{P} \), since solving even monotone integer programs on constraints with up to two variables per inequality is NP-hard, [Lag85]. (The pseudopolynomial run time of the algorithm for monotone constraints in [HN94] indicates that the problem is weakly NP-hard.)

The algorithm for solving a general SM2 is to transform such problem to a monotone SM2, a process we refer to as monotonizing, and then transform it to a problem with binary coefficients, referred to as binarizing. Notice that a problem can be monotonized first, and then binarized, or the other way around – these two processes are commutative. The technique of binarizing a monotone system of constraints was introduced by Hochbaum and Naor in [HN94]. The concept of monotonizing was introduced in [HMNT93]. Given a monotone system of constraints, the binarizing process constructs a closure graph. A minimum SM-closed set in that graph is also an optimal solution to the SM-integer minimization on that set of constraints.

5.1 Binarizing

The process of binarizing transforms the constraints into an equivalent set of MIN-2SAT constraint. We replace the \( n \) original variables and \( m \) original constraints by \( \bar{u} = \sum_{j=1}^{n} u_j \) new variables and at most \( mU + \bar{u} \) new constraints, where \( U = \max_i u_i \).

Each variable \( x_i \) is substituted by \( u_i \) binary variables \( x_{i\ell} \) (\( \ell = 1, \ldots, u_i \)), and the added constraints \( x_{i\ell} \geq x_{i,\ell+1} \) (\( \ell = 1, \ldots, u_i - 1 \)). Subject to these constraints, the correspondence between \( x_i \) and the \( u_i \)-tuple \( (x_{i1}, \ldots, x_{iu_i}) \) is one-to-one and is characterized by \( x_{i\ell} = 1 \) if and only if \( x_i \geq \ell \) (\( \ell = 1, \ldots, u_i \)), or, equivalently, \( x_i = \sum_{\ell=1}^{u_i} x_{i\ell} \). This construction is then represented as a part of a closure graph, where for each variable \( x_i \), there is a chain of infinite capacity arcs from the node representing \( x_{i,\ell+1} \) to the one representing \( x_{i\ell} \).

We now explain how to transform the constraints of the given system into constraints in terms of the binary variables \( x_{i\ell} \)’s. For a general constraint of the form, \( a_{ki}x_i + a_{kj}x_j \geq b_k \), consider the case where both are positive, and assume without loss of generality that \( 0 < b_k < a_{ki}u_i + a_{kj}u_j \). The other cases where one is negative (and the constraint is monotone), or both are negative, are similarly binarized.
For every $\ell$ ($\ell = 0, \ldots, u_i$), let $\alpha_{k\ell} = \left\lceil \frac{b_k - r_{k\ell}}{a_{kj}} \right\rceil - 1$. For any integer solution $x$, $a_{ki}x_i + a_{kj}x_j \geq b_k$ if and only if for every $\ell$ ($\ell = 0, \ldots, u_i - 1$), either $x_i > \ell$ or $x_j > \alpha_{k\ell}$.

or, equivalently, either $x_i \geq \ell + 1$ or $x_j \geq \alpha_{k\ell} + 1$.

which can be written as $x_{i,\ell+1} + x_{j,\alpha_{k\ell}+1} \geq 1$.

Obviously, if $\alpha_{k\ell} \geq u_j$, then we fix the variable, $x_{i,\ell+1} = 0$.

If the above transformation is applied to a monotone system of inequalities, then the resulting 2-SAT integer program is also monotone. Thus, altogether we have replaced one original constraint on $x_i$ and $x_j$ by at most $u_i + 1$ constraints on the variables $x_{i,\ell}$ and $x_{j,\ell}$. The other cases, corresponding to different sign combinations of $a_{ki}$, $a_{kj}$, and $b_k$, can be handled in a similar way. We thus showed,

**Theorem 2** The set of SM2 constraints is equivalent to the constraints of SM-MIN-2SAT on at most $nU$ binary variables and $mU + \bar{u}$ constraints.

### 5.2 Monotonizing and constructing an equivalent SM-closure problem

With Theorem 2 we can now assume that the SM2 problem is given as SM-MIN-2SAT problem. The only constraints in SM-MIN-2SAT (other than the binary requirements) are of the form $x_i + x_j \geq 1$ or $x_i + x_j \leq 1$ or $x_i - x_j \geq 0$.

We replace each variable $x$ by two variables, $x^+ \in \{0, 1\}$ and $x^- \in \{-1, 0\}$. The nonmonotone inequality $x_i + x_j \geq 1$ is then replaced by two monotone inequalities:

$$x_i^+ - x_j^- \geq 1$$

$$-x_i^- + x_j^+ \geq 1.$$  

The inequality $-x_i - x_j \leq 1$ is replaced by,

$$-x_i^+ + x_j^- \leq 1$$

$$x_i^- - x_j^+ \leq 1.$$  

And a monotone inequality, $x_i - x_j \geq 0$ is replaced by:

$$x_i^+ - x_j^+ \geq 0$$

$$-x_i^- + x_j^- \geq 0.$$  

We now construct a closure graph for solving the monotone problem: There is a node for each variable, and the nodes corresponding to variables in $x^+$ form the set $V^+$ and those corresponding to variables in $x^-$ form the set $V^-$. A node in $V^+$ or $V^-$ is selected in the closed set if and only if the corresponding value is $x_i^+ = 1$ or $x_j^- = 0$, respectively. For a constraint $x_i^+ - x_j^+ \geq 1$ there is an arc of infinite capacity $(x_j^-, x_i^+)$. This ensures that if $x_j^- = 0$ and $x_i^+$ is in the
closed set, then so is $x_j^+$, and thus $x_j^+ = 1$. Similarly we construct for constraint $x_i^+ - x_j^- \leq 1$ infinite capacity arc $(x_i^+, x_j^-)$, and for $x_i^+ - x_j^+ \geq 0$ an infinite capacity arc $(x_j^+, x_i^-)$. We thus proved:

**Lemma 1.** The set $\{ j \in V^+ | x_j^+ = 1 \} \cup \{ j \in V^- | x_j^- = 0 \}$ is closed in the constructed graph if and only if the solution $x^+$ and $x^-$ is feasible for the set of monotone inequalities.

With the notation $X^+ = \{ i | x_i^+ = 1 \}$ and $X^- = \{ i | x_i^- = -1 \}$ we set the objective function of this relaxed (monotonized binarized) SM2 to $\min f(X^+) + f(X^-)$ where the sets $X^+$ and $X^-$ are subsets of the respective copies of $V$, $V^+$ and $V^-$. The closed set constraints correspond to the requirement that $D_X = X^+ \cup (V^- \setminus X^-)$ is closed. Therefore, solving the monotonized problem for $x^+, x^-$ is equivalent to solving the respective SM-cut problem,

$$\min_{D_X \subseteq V, D_X \text{ closed}} f(D_X \cap V^+) + f(D_X \cap V^-)$$

Notice that this problem is equivalent to solving the min SM-cut in the reverse graph, or, we could have defined a closed set as a set containing all its predecessors. We conclude that we can solve in polynomial time the relaxed SM2 with the objective $g(X^+, X^-) = f(X^+) + f(X^-)$.

Let $X'^+ \subseteq V^+$ and $X'^- \subseteq V^-$ be the sets minimizing $g()$ among all feasible pairs of sets for the relaxed SM2. Let $S^*$ be an optimal set minimizing the function $f()$ in the (original) SM2 formulation with $S'^+$ and $S'^-$ the copies of $S^*$ in $V^+$ and $V^-$ respectively. Then,

$$2f(S^*) = f(S'^+) + f(S'^-) \geq g(X'^+ \cup X'^-) = f(X'^+) + f(X'^-) \geq f(X'^+ \cup X'^-) + f(X'^+ \cap X'^-) \geq f(X'^+ \cup X'^-).$$

The first inequality holds since $X'^+ \cup X'^-$ is an optimal solution to the relaxed SM2. The second inequality follows from the submodularity of the function $f$. $f(X'^+ \cup X'^-)$ is the value of our solution where an element is included if either one of its two copies is in $X'^+$ or in $X'^-$.

If both nodes $x_j^+$, $x_j^-$ are of value 1 and $-1$ respectively, then we set the value of $x_j = 1$. Let $V^\perp$ be the set of such variables. If both $x_j^+$, $x_j^-$ are of value 0, we let $x_j = 0$, and $V^0$ is the set of these variables. The set of remaining variables that have exactly one of $x_j^+$, $x_j^-$ of absolute value 1 and the other 0, we call $V^{\pm}$. Since the binarized problem, which is a 2SAT problem, is equivalent to the original SM2 (rather than being a relaxation of it), it is possible to find a feasible solution to the respective 2SAT expression by using the linear time algorithm of [EIS76]. Let such a feasible solution be $z_i$. The proof of Theorem 3 demonstrates that setting $x_i = z_i$ for each $i \in V^{\pm}$ is feasible for SM2.

**Theorem 3 ([HMNT93])** If the integer problem on 2 variables per inequality has a feasible solution, then the solution to the binarized monotonized constraints has a feasible rounding that is found in linear time in the number of constraints.
Let \( Z = \{ i \mid z_i = 1, i \in V^{\frac{1}{2}} \} \). Since \( V^1 \cup V^{\frac{1}{2}} = X^{t+} \cup X^{t-} \) and \( Z \subseteq V^{\frac{1}{2}} \), it follows from the monotonicity of function \( f \) that,

\[
f(V^1 \cup Z) \leq f(X^{t+} \cup X^{t-}).
\]

Therefore we conclude that \( f(V^1 \cup Z) \leq 2f(S^*) \) thus demonstrating a polynomial time 2-approximation algorithm for SM2.

### 5.3 Persistency and proof of best possible approximations

We consider the variables that are in the solution set of the relaxed SM2, \( V^1 \) and \( V^0 \) to be “integral”. The variables that are in \( V^{\frac{1}{2}} \) are considered half integral.

The proof of Theorem 3 demonstrates that the half integral solution has the persistency property. That is, if any variable \( x_i \) is integer, meaning it is in \( V^1 \) or \( V^0 \), then \( x_i \) retains this integer value in an optimal solution.

In [Hoc97], (page 132) we show that any 2-SAT is equivalent to a vertex cover problem. Therefore the impossibility of approximating SM-vertex cover in polynomial time within a factor better than 2, unless \( \text{NP}=\text{P} \), [GKTW09], implies that the 2-approximation algorithms given here for SM2 are best possible and cannot be improved.

### 6 Conclusions

We demonstrate here best possible 2-approximation algorithms for a large family of submodular optimization that are NP-hard. The technique used for all these problems is unified for all the problems as submodular minimization over linear constraints with at most two variables per inequality. The running time of the algorithms is strongly polynomial time. We further introduce here a new submodular optimization problem – the submodular-closure problem which is the foundation of all approximation algorithms for the submodular optimization over constraints with at most two variables per inequality. The results extend to multi-sets in a manner analogous to the rounding of solutions for integer problems given e.g. in [Hoc97] Section 3.8.2. This settles, for the first time, the approximation and complexity status of a number of submodular minimization problems including: SM-2SAT; SM-min satisfiability; SM-edge deletion for clique and SM-node deletion for biclique.

### References

[BR01] R. Bar-Yehuda and D. Rawitz. Efficient algorithms for integer programs with two variables per constraint. *Algorithmica* 29(4) 595–609, 2001.

[DS02] I. Dinur and S. Safra. Approximating vertex cover within a factor better than 1.36 is NP-hard. *STOC* 33–42, 2002.

[EIS76] S. Even, A. Itai and A. Shamir. On the complexity of timetable and multi-commodity flow problems. *SIAM J. Comput.* 5, 691-703, 1976.
[GKTW09] G. Goel, C. Karande, P. Tripathi and L. Wang. Approximability of Combinatorial Problems with Multi-agent Submodular Cost Functions. FOCS 2009.

[GLS88] M. Grötschel, L. Lovász, A. Schrijver. Geometric Algorithms and Combinatorial Optimization Springer-Verlag, Berlin, 1988.

[Ha96] J. Hästad. Clique is hard to approximate within $n^{1-\epsilon}$. In Proc. 37th IEEE Symp. on Foundations of Computer Science, 627–636, 1996.

[HMNT93] D. S. Hochbaum, N. Megiddo, J. Naor and A. Tamir. Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality. Math. Programming, 62, 69–83, 1993.

[HN94] D. S. Hochbaum and J. Naor. Simple and fast algorithms for linear and integer programs with two variables per inequality. SIAM Journal on Computing, 23(6) 1179–1192, 1994.

[Hoc82] D. S. Hochbaum. "Approximation Algorithms for the weighted set covering and node cover problems," SIAM Journal of Computing, 11.3 (1982), 555–556. Also, GSA Working Paper No. 64-79-80, April 1980.

[HQ03] D. S. Hochbaum and M. Queyranne. Minimizing a convex cost closure set. SIAM Journal of Discrete Math, 16:2, 192–207, 2003. An extended abstract in Lecture Notes in Computer Science 1879. M. Paterson (ed.) 8th Annual European Symposium Algorithms - ESA 2000, 256–267.

[Hoc83] D. S. Hochbaum. Efficient Bounds for the stable set, vertex cover and set packing problems. Discrete Applied Math., 6, 243–254, 1983.

[Hoc97] D. S. Hochbaum. Approximating covering and packing problems: set cover, vertex cover, independent set and related problems. Chapter 3 in "Approximation algorithms for NP-hard problems" edited by D. S. Hochbaum. PWS Boston.,1997, 94–143.

[Hoc98] D. S. Hochbaum. Approximating clique and biclique problems. Journal of Algorithms, 29, 174–200, 1998.

[Hoc02] D. S. Hochbaum. Solving integer programs over monotone inequalities in three variables: A framework for half integrality and good approximations. European J. of Operational Research 140(2), 291–321, 2002.

[IFF01] S. Iwata, L. Fleischer and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. J. of the ACM 48(4), 761–777, 2001.

[IN09] S. Iwata and K. Nagano. Submodular function minimization under covering constraints. FOCS 2009.

[IO09] S. Iwata and J. B. Orlin. A Simple Combinatorial Algorithm for Submodular Function Minimization. SODA, 1230–1237, 2009.

[KKM94] R. Kohli, R. Krishnamurti and P. Mirchandani. The minimum satisfiability problem. SIAM J. Discrete Mathematics, 7(2), 275–283, 1994.

[KY09] C. Koufogiannakis and N. Young. Greedy delta-approximation algorithm for covering with arbitrary constraints and submodular cost. ICALP 2009, Part I, LNCS 5555, 634–652, 2009.

[Lag85] J. C. Lagarias. The computational complexity of simultaneous diophantine approximation problems. SIAM Journal on Computing 14, 1985, 196–209.

[MR96] M. V. Marathe and S. S. Ravi. On approximation algorithms for the minimum satisfiability problem. Information Processing Letters, 58(1), 23–29, 1996.

[Or09] J. B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. Math. Programming, 118(2), 237–251, 2009.
[Sch00] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *J. Combin. Theory B*, 80 346-355, 2000.