SIAM J. DISCRETE MATH.
Vol. 27, No. 2, pp. 1123–1145

ALGORITHMS FOR SYMMETRIC SUBMODULAR FUNCTION MINIMIZATION UNDER HEREDITARY CONSTRAINTS AND GENERALIZATIONS*

MICHEL X. GOEMANS† AND JOSÉ A. SOTO‡

Abstract. We present an efficient algorithm to find nonempty minimizers of a symmetric submodular function f over any family of sets $\mathcal{I}$ closed under inclusion. Our algorithm makes $O(n^3)$ oracle calls to f and $\mathcal{I}$, where n is the cardinality of the ground set. In contrast, the problem of minimizing a general submodular function under a cardinality constraint is known to be inapproximable within $o(\sqrt{n/\log n})$ [Z. Svitkina and L. Fleischer, *in Proceedings of the* 49*th Annual IEEE Symposium on Foundations of Computer Science*, IEEE, Washington, DC, 2008, pp. 697–706]. We also present two extensions of the above algorithm. The first extension reports all nontrivial inclusionwise minimal minimizers of f over $\mathcal{I}$ using $O(n^3)$ oracle calls, and the second reports all extreme subsets of f using $O(n^4)$ oracle calls. Our algorithms are similar to a procedure by Nagamochi and Ibaraki [*Inform. Process. Lett.*, 67 (1998), pp. 239–244] that finds all nontrivial inclusionwise minimal minimizers of a symmetric submodular function over a set of size n using $O(n^3)$ oracle calls. Their procedure in turn is based on Queyranne's algorithm [M. Queyranne, *Math. Program.*, 82 (1998), pp. 3–12] to minimize a symmetric submodular function by finding *pendent pairs*. Our results extend to any class of functions for which we can find a pendent pair whose head is not a given element.

Key words. submodular functions, symmetric, minimum cut

AMS subject classifications. 90C27, 68Q25

DOI. 10.1137/120891502

1. Introduction.

A real valued function f is called a *set function on* V if its domain consists of all subsets of a finite set V. We assume that f is given through a *value oracle* which, for any input $S \subseteq V$, returns $f(S)$. A set function $f\colon 2^V \to \mathbb{R}$ is called *submodular* over V if

$$f(A \cup B) + f(A \cap B) \leq f(A) + f(B) \tag{1.1}$$

for every pair of subsets A and B of V. The function f is further called *symmetric* if

$$f(A) = f(V \setminus A) \text{ for all } A \subseteq V. \tag{1.2}$$

Submodularity is observed in a wide family of problems. The rank function of a matroid, the cut function of a (nonnegatively weighted, directed, or undirected) graph, the entropy of a set of random variables, and the logarithm of the volume of the parallelepiped formed by a set of vectors are all examples of submodular functions. Many combinatorial optimization problems can be formulated as minimizing a submodular function; this is, for example, the case for the problem of finding the smallest number of edges to add to make a graph k-edge-connected. Therefore, the following problem is considered fundamental in combinatorial optimization.

*Received by the editors September 14, 2012; accepted for publication (in revised form) March 7, 2013; published electronically June 25, 2013. This work was partially supported by NSF contracts CCF-0829878 and CCF-1115849 and by ONR grant N00014-11-1-0053.
http://www.siam.org/journals/sidma/27-2/89150.html

†Department of Mathematics, MIT, Cambridge, MA 02139 (goemans@math.mit.edu).
‡DIM-CMM, University of Chile, Santiago, Chile and Technische Universität Berlin, Berlin, Germany (jsoto@dim.uchile.cl). This author was partially supported by Núcleo Milenio Información y Coordinación en Redes ICM/FIC P10-024F.

Submodular function minimization problem. Given a submodular function $f\colon 2^V \to \mathbb{R}$, find a subset $X^* \subseteq V$ that minimizes $f(X^*)$.

Grötschel, Lovász, and Schrijver [6, 7] show that the above problem can be solved using the ellipsoid method in strongly polynomial time and using a polynomial number of oracle calls. Later, a series of combinatorial strongly polynomial algorithms, starting with the works of Iwata, Fleischer, and Fujishige [8] and of Schrijver [23], was developed [2, 8, 11, 18, 23, 10]. The current fastest combinatorial algorithm known, due to Orlin [18], runs in $O(n^5 \log n)$ time and makes $O(n^5)$ function oracle calls, where n is the size of the ground set.

Faster algorithms are available when the function f has more structure. The case where f is symmetric is of special interest. In this case we also require the minimizer X^* of f to be a *nontrivial* subset of V, that is, $\emptyset \subset X^* \subset V$; otherwise the problem becomes trivial since, by symmetry and submodularity, $f(\emptyset) = \frac{1}{2}(f(\emptyset) + f(V)) \leq \frac{1}{2}(f(X) + f(V \setminus X)) = f(X)$ for all $X \subseteq V$.

The canonical example of a symmetric submodular function (SSF) is the cut function of a nonnegatively weighted undirected graph. Minimizing such a function corresponds to the minimum cut problem. Nagamochi and Ibaraki [13, 14] give a combinatorial algorithm to solve this problem without relying on network flows. This algorithm has been improved and simplified independently by Stoer and Wagner [24] and Frank [3]. Queyranne [20] generalizes it and obtains a purely combinatorial algorithm that minimizes any SSF using only $O(|V|^3)$ function oracle calls.

In this paper, we focus on the problem of minimizing set functions over subfamilies of 2^V that are closed under inclusion. More precisely, a *hereditary family* $\mathcal{I}$ (also called a lower ideal or a down-monotone family) over V is a collection of subsets of V such that if a set is in the family, so are all its subsets. A natural set function minimization problem is the following.

Hereditary set function minimization problem. Given a set function f on V and a hereditary family $\mathcal{I}$ over V, find a subset $\emptyset \neq X^* \in \mathcal{I}$ that minimizes $f(X)$ over all sets $X \in \mathcal{I} \setminus \{\emptyset\}$.

Common examples of hereditary families over V include the following.

- *Cardinality families.* For $k \geq 0$, consider the family of all subsets with at most k elements, $\mathcal{I} = \{A \subseteq V : |A| \leq k\}$.
- *Knapsack families.* Given a weight function $w\colon V \to \mathbb{R}_+$, consider the family of all subsets of weight at most one unit, $\mathcal{I} = \{A \subseteq V : \sum_{v \in A} w(v) \leq 1\}$.
- *Matroid families.* The family of independent sets of a matroid with ground set V.
- *Hereditary graph families.* Given a graph $G = (V, E)$, consider the family of sets S of vertices such that the induced subgraph $G[S]$ satisfies a certain hereditary property such as being a clique, being triangle-free, being planar, or excluding certain minors.
- *Matching families.* Given a hypergraph H with edge set V, consider the family of matchings of H, i.e., sets of pairwise disjoint edges.

We assume that the hereditary family $\mathcal{I}$ is given through a *membership oracle* which, for any input $S \subseteq V$, reports whether $S \in \mathcal{I}$ or not. Noting that the intersection of hereditary families is also hereditary, we can see that the minimization problem defined above is very general. In fact, when f is a general (nonsymmetric) submodular function, this problem cannot be approximated within $o(\sqrt{|V|/\log|V|})$ using a polynomial number of oracle calls even for the simple case of cardinality families (see [25]). For the case where f is symmetric, we extend Queyranne's algorithm as follows.

THEOREM 1. *Given an SSF f on V and a hereditary family $\mathcal{I}$ over V, an optimal solution for the associated hereditary minimization problem can be found in $O(|V|^3)$ time and using $O(|V|^3)$ oracle calls to f and $\mathcal{I}$.*

In this statement, an *optimal solution* refers to a nonempty set $X^* \in \mathcal{I}$ that attains the minimum in the hereditary minimization problem. Our algorithm in fact returns a *minimal* solution among all optimal solutions, that is, one such that no proper subset of it is also optimal.

The result above implies, for example, polynomial time algorithms for the following problems:

1. Find a minimum *unbalanced* cut in a graph; that is, for given k, find among all nonempty sets of at most k vertices the one inducing a minimum cut.
2. More generally, given a nonnegatively weighted graph, find a nonempty induced subgraph satisfying a hereditary graph property (e.g. triangle-free, clique, stable-set, or planar) minimizing the weights of the edges having precisely one endpoint in the subgraph.

For the unconstrained SSF minimization problem, Nagamochi and Ibaraki [15] present a modification of Queyranne's algorithm that finds all inclusionwise minimal minimizers of an SSF still using a cubic number of oracle calls. Using similar ideas, we can also list all minimal solutions of a hereditary minimization problem using only $O(|V|^3)$ oracle calls. As these minimal solutions can be shown to be disjoint, there are at most $|V|$ of them. More precisely, we show the following theorem.

THEOREM 2. *Given an SSF f on V and a hereditary family $\mathcal{I}$ over V, the collection of all minimal optimal solutions of the associated hereditary minimization problem can be found in $O(|V|^3)$ time and using $O(|V|^3)$ oracle calls to f and $\mathcal{I}$.*

Our methods can be used to solve the hereditary minimization problem on a more general class of functions that we denote as *strongly PP-admissible functions*. These are the functions f such that every one of its fusions admits a polynomial time procedure returning a so-called *pendent pair* whose *head* avoids a given element. See sections 2 and 4 for definitions and precise statements. The works of Queyranne [20], Nagamochi and Ibaraki [15], and Rizzi [21] show that the class of strongly PP-admissible functions includes crossing SSFs, their *restrictions* (these are functions that are both *intersecting submodular* and *intersecting posimodular*), and the case where $f(S)$ is defined as $d(S, V \setminus S)$ for a monotone and consistent symmetric set map d in the sense of Rizzi. An example of the latter setting is to find an induced subgraph $G[S]$ satisfying a certain hereditary property (e.g., being planar or bipartite) and minimizing the maximum (weighted) distance between any vertex in S and any vertex in $V \setminus S$ (note that this does not define a submodular function). For all the functions mentioned in this paragraph, it is possible to find a pendent pair whose head avoids a single element by constructing a so-called *maximum adjacency order*.

Finally, we show how to extend our methods to return all the *extreme subsets* of certain classes of set functions, where a set is called extreme if its function value is strictly smaller than any one of its nontrivial subsets. In particular, the minimal minimizers of a set function over any hereditary family are extreme subsets. For our methods to work, we further require the family of extreme subsets of f to form a simple structure known as a *laminar family*. This is the case for most of the strongly PP-admissible functions we consider. The general versions of our results about minimal minimizers and extreme sets are stated as Theorems 18 and 24 in sections 5 and 6, respectively.

Related work. Constrained submodular function minimization problems, i.e., the minimization of a submodular function over *subfamilies* of 2^V, have been studied in different contexts. Padberg and Rao [19] show that the minimum odd cut problem, obtained by restricting the minimization over all odd sets, can be solved in polynomial time. This was generalized to submodular functions over larger families of sets (satisfying certain axioms) by Grötschel, Lovász, and Schrijver [7] and by Goemans and Ramakrishnan [5]. This covers, for example, the minimization over all even sets, or all sets not belonging to a given antichain, or all sets excluding all minimizers (i.e., to find the second minimum). For the particular case of minimizing a *symmetric* submodular function under cardinality constraints the best previous result is a 2-approximation algorithm by Dughmi [1]. Goel et al. [4] have studied the minimization of *monotone* submodular functions constrained to sets satisfying combinatorial structures on graphs, such as vertex covers, shortest paths, perfect matchings, and spanning trees, giving inapproximability results and almost matching approximation algorithms for them. Independently, Iwata and Nagano [9] study both the vertex and the edge covering versions of this problem.

The algorithm of Nagamochi and Ibaraki [15] also works with functions satisfying a less restrictive symmetry condition. Narayanan [17] shows that Queyranne's algorithm can be used to minimize a wider class of submodular functions, namely, functions that are contractions or restrictions of SSFs. Rizzi [21] has given a further extension of this algorithm for a different class of functions. Nagamochi [16] has recently given an efficient algorithm to find all extreme subsets of an SSF and some extensions. This algorithm is not based on the ability to find pendent pairs but on a different structure denoted as *flat pairs*.

Paper organization. In section 2 we define pendent pairs and revisit Queyranne's procedure for finding nontrivial minimizers of set functions admitting pendent pairs. This algorithm solves the unconstrained problem, provided we have access to a black box for finding pendent pairs, not only for the original function but also for every one of its fusions.

In section 3 we modify the procedure above to solve the hereditary minimization problem of certain set functions, denoted as *strongly PP-admissible functions*, provided there is a stronger black box that finds, for every fusion, a pendent pair whose head avoids any fixed element.

In section 4 we present a general class of strongly PP-admissible functions, including SSFs, for which the problem of finding a pendent pair whose head avoids an element can be solved efficiently by finding a so-called maximum adjacency order.

In section 5 we summarize our results for the hereditary minimization problem and give some extensions to cohereditary systems.

In section 6 we turn to the problem of finding extreme subsets of a set function, and we give a pendent pair based algorithm to find all of them in certain cases. We compare our results to a recent work of Nagamochi [16] that is based on a different structure known as flat pair. We conclude this article with a section of relevant examples.

2. Pendent pairs and a review of Queyranne's algorithm. We start this section by introducing some notation. A pair (V, f) where f is a set function over V is called a *(set function) system.* For any set $A \subseteq V$, any $x \in A$, and any $y \in V \setminus A$, we use the convention that, $A + y$ and $A - x$ stand for the sets $A \cup \{y\}$ and $A \setminus \{x\}$, respectively. Also, for $x \in V$ we use $f(x)$ to denote $f(\{x\})$.

Let Π be a partition of V. For every collection of parts $X \subseteq \Pi$, we use V_X to denote the set of elements contained in the union of the parts in X; this is

$$V_X = \bigcup_{S \in X} S \subseteq V. \tag{2.1}$$

The *fusion of* f *relative to* Π, denoted by f_Π, is the set function on Π given by

$$f_\Pi(X) = f(V_X). \tag{2.2}$$

We say that (V', f') is a fusion of (V, f) if V' corresponds, up to renaming, to a partition of V and f' is the fusion of f relative to this partition.

By *fusing* a collection of elements $S \subseteq V$ into a new single element s we mean replacing the system (V, f) by the system $((V \setminus S)+s, f_\Pi)$, where Π is the partition that has one part equal to S (*renamed* as s) and where all the other parts are singletons, which keep the same name as the unique element they contain.

Queyranne's technique performs iterative fusions on the original system (V, f). To keep our explanation simple, we overload the notation above by saying that for any fusion (V', f') of (V, f) and any $x \in V'$, V_x is the set of elements in the original set V that have been fused into x, and for every set $X \subseteq V'$, V_X is the union of all sets V_x with $x \in X$. Furthermore, we say that a set $A \subseteq V$ is *present* in a fused system (V', f') if there is a set $B \subseteq V'$ such that $A = V_B$.

A set $X \subseteq V$ *separates* two elements t and u of V if X contains exactly one of t and u. We extend this notion to fusions by saying that two elements t and u in V' are separated by a set $X \subseteq V$ if $V_t \subseteq X$ and $V_u \subseteq V \setminus X$ or vice versa.

The following concept is crucial for the development of Queyranne's technique. An ordered pair (t, u) of different elements of V is called a *pendent pair* of the system (V, f) if $\{u\}$ has the minimum f-value among all the subsets of V separating u and t; this is

$$f(u) = \min\{f(U) \colon U \subset V, |U \cap \{t, u\}| = 1\}. \tag{2.3}$$

The element u is called the *head* of the pendent pair (t, u). We say that the system (V, f) is *PP-admissible* if for every fusion (V', f') with $|V'| \geq 2$ there exists a pendent pair (t, u) for (V', f'). Observe that by definition, PP-admissibility is closed under taking fusions.

Suppose that (V, f) is a PP-admissible system and that (t, u) is a pendent pair for (V, f). Let X^* be a nontrivial minimizer of (V, f). Then we have two cases depending on whether X^* separates t and u or not. If X^* separates t and u, then by the definition of a pendent pair, $f(u) \leq f(X^*)$, and so $\{u\}$ is also a nontrivial minimizer. If this is not the case, consider the set system (V', f') obtained by fusing t and u into a single element tu. Any nontrivial minimizer X' of this system induces a nontrivial minimizer $V_{X'}$ of (V, f).

By iteratively applying the above argument $n - 1$ times (as all the fused systems admit pendent pairs) we can find a nontrivial minimizer of (V, f) as the set having minimum value among all sets V_u, where (t, u) is the pendent pair found in a given iteration. The procedure is described as Algorithm 1, which we call *Queyranne's routine*.

The above argument shows that Queyranne's routine is correct. The only nontrivial step of the routine corresponds to finding pendent pairs. Suppose in what follows that we have access to a black box algorithm $\mathcal{A}$ that computes pendent pairs

Algorithm 1. (Queyranne's routine).

Input: A PP-admissible system (V, f).
Output: A nontrivial minimizer X^* for (V, f).

```
(V', f') ← (V, f) and C ← ∅.                ▷ C is the set of candidates for minimum.
while |V'| ≥ 2 do
    Find any pendent pair (t, u) for (V', f').
    Add V_u to C.        ▷ V_u is the set of elements of V that have been fused into u.
    Update (V', f') by fusing {t, u} into a single element tu.
end while                                                              ▷ |V'| = 1.
Return a set X* in C with minimum f-value.
```

for any fusion of (V, f) in $O(T(|V|))$ time and using $O(T(|V|))$ calls to a value oracle for some function $T(\cdot)$.

Lemma 3. *By using $\mathcal{A}$ as a subroutine, Queyranne's routine returns a nontrivial minimizer of (V, f) in $O(|V| \cdot T(|V|))$ time and using the same asymptotic number of oracle calls.*

Queyranne [20] originally devised the routine above for SSFs. He shows not only that these functions are PP-admissible, a fact originally shown by Mader [12], but also that we can compute pendent pairs for a system (V, f) in time $O(|V|^2)$ and using $O(|V|^2)$ oracle calls. In section 4 we discuss how to perform this and give some extensions.

By using the lemma above and the fact that SSFs are closed for fusions, Queyranne has shown the following theorem.

Theorem 4 (Queyranne [20]). *The problem of finding a nontrivial minimizer of an SSF f over V can be solved in $O(|V|^3)$ time and using $O(|V|^3)$ oracle calls to f.*

In section 3, we extend Queyranne's algorithm to the problem of finding nontrivial minimizers of certain set functions under hereditary constraints.

3. Hereditary minimization problem.

A triple $(V, f, \mathcal{I})$, where f is a set function over V and $\mathcal{I}$ is a hereditary family of V, is called a *hereditary system*. A set X^* is an *optimal* solution for $(V, f, \mathcal{I})$ if X^* is a minimizer of the function f over the nonempty sets in $\mathcal{I}$. The set X^* is a *minimal optimal* solution for $(V, f, \mathcal{I})$ if X^* is a (nonempty) inclusionwise minimal optimal solution for the hereditary system. We also say that X^* is a minimal optimal solution for (V, f) if X^* is minimal optimal for $(V, f, 2^V)$.

We extend the notion of fusion to hereditary systems as follows. Given a partition Π of V, the *fusion of $\mathcal{I}$ relative to* Π, denoted by $\mathcal{I}_\Pi$, is the family

$$\mathcal{I}_\Pi = \{I \subseteq \Pi \colon V_I \in \mathcal{I}\}. \tag{3.1}$$

It is easy to see that if $\mathcal{I}$ is hereditary, then so is $\mathcal{I}_\Pi$. We say that $(V', f', \mathcal{I}')$ is a fusion of $(V, f, \mathcal{I})$ if this system is, up to renaming of the elements, equal to $(V_\Pi, f_\Pi, \mathcal{I}_\Pi)$ for some partition Π of V.

It is worth noting that if $(V', f', \mathcal{I}')$ is a specific fusion of $(V, f, \mathcal{I})$, then, for every $A \subseteq V'$, we can test if $A \in \mathcal{I}'$ and we can evaluate $f'(A)$ using only one oracle call to $\mathcal{I}$ and f, respectively.

In order to find optimal solutions of hereditary systems we require a stronger admissibility condition. We say that a system (V, f) is *strongly PP-admissible* if, for every fusion (V', f') where V' has at least three elements, and every $s \in V'$, there is a pendent pair (t, u) for (V', f') whose head avoids s. By this we mean that $u \neq s$. We say that $(V, f, \mathcal{I})$ is strongly admissible when (V, f) is. Strong PP-admissibility is closed under taking fusions. Observe also that strongly PP-admissible functions are not necessarily symmetric. In fact, symmetry is equivalent to the property of admitting pendent pairs avoiding a fixed element for fusions in which the partition V' has exactly two elements (in the definition of strong PP-admissibility we require that $|V'| \geq 3$).

The lemmas in this section are useful for our purposes and may be of interest on their own.

LEMMA 5. *Let U be a nonsingleton minimal optimal solution for a hereditary system $(V, f, \mathcal{I})$. If (t, u) is a pendent pair of (V, f) and $u \in U$, then $t \in U$.*

Proof. Suppose by contradiction that $t \notin U$. Since (t, u) is a pendent pair and U separates t and u, $f(u) \leq f(U)$. This contradicts the minimality of U because $\{u\}$ is a proper subset of U. ∎

LEMMA 6. *The minimal optimal solutions of a strongly PP-admissible system $(V, f, \mathcal{I})$ are pairwise disjoint.*

Proof. Assume that A and B are two nondisjoint minimal optimal solutions. Since no one includes the other, the sets $A \setminus B$, $B \setminus A$, and $A \cap B$ are nonempty. We have two cases: either $A \cup B = V$, or $V \setminus (A \cup B) \neq \emptyset$.

For the first and second cases, respectively, consider the systems $(V' = \{a, b, c\}, f', \mathcal{I}')$ and $(V' = \{a, b, c, d\}, f', \mathcal{I}')$ obtained by fusing $A \setminus B$ into a, $B \setminus A$ into b, $A \cap B$ into c, and, only for the second case, $V \setminus (A \cup B)$ into d. In both systems, $V_{\{a,c\}} = A$ and $V_{\{b,c\}} = B$, and so the sets $\{a, c\}$ and $\{b, c\}$ are minimal optimal solutions.

We claim that there is no pendent pair (t, u) for (V', f') with $u \neq d$ (in the first case, this means that there is no pendent pair at all). The validity of this claim contradicts the strong PP-admissibility of (V, f) and completes the proof.

To prove the claim suppose that (t, u) is a pendent pair with $u \neq d$. Note that $u \neq c$, for if they were equal, Lemma 5 would imply that t is in both $\{a, c\}$ and $\{b, c\}$, which is a contradiction.

Then, without loss of generality, we can assume that $u = a$. In this case, Lemma 5 implies that $t = c$. But then, as (c, a) is a pendent pair, we have $f'(a) \leq f'(\{b, c\})$ or, equivalently, $f(A \setminus B) \leq f(B) = f(A)$, contradicting the minimality of A and completing the proof. ∎

In what follows define $\mathcal{M} = \mathcal{M}(V, f, \mathcal{I})$ as the family of minimal minimizers of a strongly PP-admissible system $(V, f, \mathcal{I})$. Define also the *minimal partition* $\overline{\mathcal{M}} = \overline{\mathcal{M}}(V, f, \mathcal{I})$ of V whose parts are all the sets in $\mathcal{M}$ (which are disjoint by the previous lemma), plus possibly one extra part $V \setminus \bigcup_{X \in \mathcal{M}} X$, which we call the *bad* part of $\overline{\mathcal{M}}$. The following lemma relates the pendent pairs of (V, f) with the minimal partition $\overline{\mathcal{M}}$.

LEMMA 7. *Let (t, u) be a pendent pair of (V, f). At least one of the following holds:*

(i) *t and u are in the same part of $\overline{\mathcal{M}}$.*

(ii) *u is a* loop[1] *of $\mathcal{I}$.*

(iii) *$\{u\} \in \mathcal{M}$.*

Proof. Suppose that (i) does not hold. Then there are different parts of $\overline{\mathcal{M}}$, say, T and U, such that $t \in T$ and $u \in U$. If U is a part in $\mathcal{M}$, then, by Lemma 5, U must be a singleton, and so (iii) holds. So assume that U is the bad part of $\overline{\mathcal{M}}$. Then T is a minimal optimal solution separating t and u. Since (t, u) is a pendent pair, we conclude that $f(u) \leq f(T)$. But then u has to be a loop, as otherwise $\{u\}$ would be a minimal optimal solution strictly contained in U. ∎

[1] A loop u of a hereditary family $\mathcal{I}$ is an element such that $\{u\}$ is not in $\mathcal{I}$.

In what follows we present two algorithms: one to find a particular minimal optimal solution of a strongly PP-admissible system $(V, f, \mathcal{I})$ and another to find all of them. To simplify our discussion we assume that $\mathcal{I}$ is not trivial (in other words, $\mathcal{I} \notin \{\emptyset, \{\emptyset\}, 2^V\}$).

If we simply use Queyranne's routine on (V, f), we could consider candidates that are not in the hereditary family. In order to avoid that, we make two changes. First, we impose that the (fused) system considered in each iteration has at most one loop. We do this by fusing all the loops (if any) of the current system into a single loop that we denote as s. The second change is that the pendent pair (t, u) we use in every iteration must satisfy $u \neq s$ (this is possible by strong admissibility).

As in Queyranne's routine, we add the set V_u associated to the head u to the family $\mathcal{C}$ of candidates, and then we fuse t and u together. Since u is not a loop, the candidate V_u is in the hereditary family $\mathcal{I}$. At the end of the procedure, we return the candidate having minimum f-value. If there are several with the same value, we return the one that was considered earlier. As we show below, this set is a minimal optimal solution.

The complete procedure, denoted as FindMinimal, is depicted below as Algorithm 2. We defer the problem of finding pendent pairs to section 4. For now assume that we have access to a black box algorithm $\mathcal{A}$ that computes a pendent pair avoiding an element on any given fusion of (V, f) of size at least three in $O(T(|V|))$ time and using $O(T(|V|))$ oracle calls to f and $\mathcal{I}$ for some function $T(\cdot)$.

Algorithm 2. FindMinimal $(V, f, \mathcal{I})$.

Input: A strongly admissible hereditary system $(V, f, \mathcal{I})$, where $\mathcal{I}$ is not trivial.
Output: A minimal optimal set X^* for the hereditary minimization problem.

Set $(V', f', \mathcal{I}') \leftarrow (V, f, \mathcal{I})$, and $\mathcal{C} \leftarrow \emptyset$. $\triangleright$ $\mathcal{C}$ is the set of candidates.
while $\mathcal{I}'$ has no loops and $|V'| \geq 3$ **do**
  Find any pendent pair (t, u) of (V', f').
  Add V_u to $\mathcal{C}$. $\triangleright$ V_u is the set of elements of V that have been fused into u.
  Update $(V', f', \mathcal{I}')$ by fusing $\{t, u\}$ into a single element tu.
end while $\triangleright$ $\mathcal{I}'$ has at least one loop.
Update $(V', f', \mathcal{I}')$ by fusing all the loops of $\mathcal{I}'$ into a single element called $s \in V'$.
while $|V'| \geq 3$ **do**
  Find a pendent pair (t, u) of (V', f') with $u \neq s$.
  Add V_u to $\mathcal{C}$.
  if $\{t, u\} \in \mathcal{I}'$ **then**
    Update $(V', f', \mathcal{I}')$ by fusing $\{t, u\}$ into a single element tu.
  else
    Update $(V', f', \mathcal{I}')$ by fusing $\{s, t, u\}$ into a single element s.
  end if
end while
if $|V'| = 2$ **then**
  Add V_x to $\mathcal{C}$ for all nonloops $x \in V'$
end if
Among the sets in $\mathcal{C}$ of minimum f-value, return the set X^* that was added earlier.

Theorem 8. *By using $\mathcal{A}$ as a subroutine, Algorithm* FindMinimal *returns a minimal optimal solution of the strongly PP-admissible hereditary system $(V, f, \mathcal{I})$ in $O(|V| \cdot T(|V|))$ time and using the same asymptotic number of oracle calls.*

Proof. The running time claim follows immediately since each iteration decreases the size of V' by one or two units, and so we focus on checking correctness.

By construction, at the beginning of every iteration of both while-loops, either $\mathcal{I}'$ is loopless or s is its only loop. In particular, every set V_u added to $\mathcal{C}$ is obtained from a nonloop u, implying that every candidate is in the hereditary family $\mathcal{I}$. To conclude we show that at least one minimal optimal solution is added to $\mathcal{C}$ and that every candidate considered before it is not optimal.

Consider the partition $\overline{\mathcal{M}} = \overline{\mathcal{M}}(V, f, \mathcal{I})$ of V defined earlier. Recall that a set $A \subseteq V$ is said to be present in a given fusion $(V', f', \mathcal{I}')$ if there is a set $B \subseteq V'$ such that $A = V_B$. The parts of $\overline{\mathcal{M}}$ are all present in the system obtained by fusing all the loops together into one. After that, they stay present from one iteration to the next provided the elements we fuse at the end of that iteration belong to the same part[2] of $\overline{\mathcal{M}}$. We have two cases: either all the parts of $\overline{\mathcal{M}}$ survive until the end of the second while-loop, or they do not.

Since $\mathcal{I}$ is not trivial, $\overline{\mathcal{M}}$ has at least two parts. If the first case occurs, then after the second while-loop, the set V' must have exactly two elements. But then $\overline{\mathcal{M}}$ must be a partition with two parts. Both sets in $\overline{\mathcal{M}}$ are checked for addition to $\mathcal{C}$ (as one of them may be bad) in line 18. We conclude this case by noting that all candidates considered before this step are strict subsets of a set in $\overline{\mathcal{M}}$, and so they are not optimal.

For the second case, consider the last iteration in which every set of $\overline{\mathcal{M}}$ is present, and let (t, u) be the pendent pair found at that time. Observe that the candidates added to $\mathcal{C}$ before this iteration are strict subsets of parts of $\overline{\mathcal{M}}$, and so they are not optimal. To conclude, we prove that the set V_u added in this iteration is a minimal optimal solution. Indeed, if this is not the case, then $\{u\}$ is not a minimal optimal solution of $(V', f', \mathcal{I}')$. Since $u \neq s$, u is not a loop either; therefore, by Lemma 7, t and u must be in the same part of $\overline{\mathcal{M}}$.[3] But since after this iteration at least one set of $\overline{\mathcal{M}}$ stops being present, we must be in the case where s, t, and u are fused together. According to the algorithm, this fusion happens only when $\{t, u\}$ is not in $\mathcal{I}'$, meaning that all s, t, and u are in the bad part of $\overline{\mathcal{M}}$. But this is a contradiction, since this particular fusion preserves the presence of all parts of $\overline{\mathcal{M}}$. ∎

We can use that the minimal optimal solutions are disjoint to find all of them: Compute the optimal value $\lambda^* = \min_{X \in \mathcal{I}^*} f(X)$ of the original system $(V, f, \mathcal{I})$ and using FindMinimal. While V is not a singleton, find a minimal optimal solution X^* of the current system using FindMinimal, and add it to the set of solutions if its f-value is λ^*. Afterward, modify $\mathcal{I}$ by removing all the sets containing X^* from the hereditary family, and repeat. We recover the entire collection of minimal optimal solutions of $(V, f, \mathcal{I})$ in this way. A naive implementation of this procedure may require $O(|V|)$ calls to FindMinimal.

We can give a better implementation, which is similar to the algorithm presented by Nagamochi and Ibaraki [15], to find all the minimal optimal solutions in the unconstrained setting, as follows.

We start by finding a particular minimal optimal solution X^* of $(V, f, \mathcal{I})$ using FindMinimal and add it to the output family. We then check if there are *singleton* minimal optimal solutions in $\mathcal{I}$ by testing the value of all of them. Afterward, we fuse the set X^* together with all the singleton minimal optimal solutions found and all the loops of $\mathcal{I}$ into a single element s, which (if it is not already a loop) we consider

[2]By this we mean that if we fuse a set $R \subseteq V'$ together, then the set $V_R \subseteq V$ of original elements is a subset of a part in $\overline{\mathcal{M}}$.

[3]More precisely, V_t and V_u are subsets of the same part of $\overline{\mathcal{M}}$.

as a loop. The new system $(V', f', \mathcal{I}')$ contains exactly one loop, and every nonloop has value strictly larger than that of X^*.

We now proceed iteratively. In each iteration, we find a pendent pair (t, u) such that $u \neq s$ and fuse t and u together into a single element tu. We then check if the new singleton is optimal (if so, we add it to the solution and consider it as a new loop, fusing it with s), or if it is a loop (in which case, we also fuse it with s). We continue doing this until the system becomes trivial. The full implementation is depicted below as Algorithm 3, which we denote as FindAllMinimals. Similar to the case of FindMinimal, we assume the existence of a black box $\mathcal{A}$ that, given a fusion of the system (V, f) having at least three elements, and an element s, returns a pendent pair whose head avoids s in $O(T(|V|))$ time and using $O(T(|V|))$ oracle calls to f and $\mathcal{I}$ for some function $T(\cdot)$.

Algorithm 3. FindAllMinimals $(V, f, \mathcal{I})$.

Input: A strongly PP-admissible hereditary system $(V, f, \mathcal{I})$, where $\mathcal{I}$ is not trivial.
Output: The family $\mathcal{M}$ of minimal optimal solutions for the hereditary minimization problem.

1: Using FindMinimal, compute a minimal optimal solution X^* of $(V, f, \mathcal{I})$.
2: Set $\mathcal{M} \leftarrow \{X^*\}$ and $\lambda^* \leftarrow f(X^*)$.
3: Add to $\mathcal{M}$ every singleton $\{v\} \in \mathcal{I}$ with $f(v) = \lambda^*$.
4: Let $(V', f', \mathcal{I}')$ be the system obtained by fusing together all the loops of $\mathcal{I}$, and the elements of all the sets added to $\mathcal{M}$. Denote the resulting new element as s.
5: $\mathcal{I}' \leftarrow \mathcal{I}' \setminus \{A \in \mathcal{I}' : s \in A\}$. ▷ If s is not a loop, consider it as one.
6: **while** $|V'| \geq 3$ **do** ▷ $f'(v) > \lambda^*$ for all $v \in V'$, and s is the only loop of $\mathcal{I}'$.
7:     Find a pendent pair (t, u) of (V', f') with $u \neq s$.
8:     **if** $\{t, u\} \in \mathcal{I}'$ and $f'(\{t, u\}) = \lambda^*$ **then**
9:         Add $V_{\{t,u\}}$ to $\mathcal{M}$.
10:         Update $(V', f', \mathcal{I}')$ by fusing $\{s, t, u\}$ into a single element s.
11:     **else if** $\{t, u\} \in \mathcal{I}'$ and $f'(\{t, u\}) > \lambda^*$ **then**
12:         Update $(V', f', \mathcal{I}')$ by fusing $\{t, u\}$ into a single element tu.
13:     **else** ▷ $\{t, u\} \notin \mathcal{I}'$.
14:         Update $(V', f', \mathcal{I}')$ by fusing $\{s, t, u\}$ into a single element s.
15:     **end if**
16: **end while**
17: Return the family $\mathcal{M}$.

Theorem 9. *By using $\mathcal{A}$ as a subroutine, Algorithm* FindAllMinimals *outputs all minimal optimal solutions of the strongly PP-admissible hereditary system* $(V, f, \mathcal{I})$ *in $O(|V| \cdot T(|V|))$ time and using the same asymptotic number of oracle calls.*

Proof. The claim about the running time follows from Theorem 8 and the fact that each iteration decreases the cardinality of V' by at least one unit. Let $\mathcal{M}_A$ be the collection returned by the algorithm and $\mathcal{M} = \mathcal{M}(V, f, \mathcal{I})$ be the family of minimal optimal solutions of $(V, f, \mathcal{I})$.

Since all the solutions added to $\mathcal{M}_A$ are disjoint and optimal by construction, it is enough to show that every minimal optimal solution X of $\mathcal{M}$ is eventually added to $\mathcal{M}_A$.

Suppose that this is not the case, and let $Y \in \mathcal{M} \setminus \mathcal{M}_A$. Let $(V_0, f_0, \mathcal{I}_0)$ be the system obtained after fusing X^*, all the singleton minimal optimal solutions, and the loops into s (see line 4). Every system considered in the algorithm after that point is a fusion of $(V_0, f_0, \mathcal{I}_0)$.

Consider the minimal partition $\overline{\mathcal{M}}_0 = \overline{\mathcal{M}}(V_0, f_0, \mathcal{I}_0)$ and observe that Y is a part of $\overline{\mathcal{M}}_0$. We claim that Y is not present at the end of the while-loop. Indeed, if it were present, then, since V' has at most two elements at that moment, and one of them is a loop, Y must have been fused into the only nonloop singleton of V'. But this is a contradiction, since the algorithm ensures that every singleton ever created is not optimal.

Consider then the last iteration in which Y is present in the fusion $(V', f', \mathcal{I}')$, and let (t, u) be the pendent pair found at that time. Since, by construction, u is neither a loop nor a singleton optimal solution of the current system, Lemma 7 implies that t and u are in the same part of the current partition $\overline{\mathcal{M}}(V', f', \mathcal{I}')$. It follows that t and u are both inside or both outside Y.[4] Since Y stops being present after this iteration, we must be in the case where t and u are inside Y, s is outside Y, and we fuse $\{s, t, u\}$ together into s. We do this only when $\{t, u\} \in \mathcal{I}'$ and $f'(\{t, u\}) = \lambda^*$ or when $\{t, u\} \notin \mathcal{I}'$. As $V_{\{t,u\}} \subseteq Y \in \mathcal{I}$, we must be in the first case. Then, according to the algorithm, $V_{\{t,u\}}$ is an optimum solution that is added to $\mathcal{M}_A$. By minimality of Y we obtain $Y = V_{\{t,u\}}$, which contradicts that $Y \notin \mathcal{M}_A$ and concludes the proof. ∎

4. Strongly PP-admissible functions.

In this section, we study different families of strongly PP-admissible set functions and show how to find pendent pairs in all of them. Most of the presented families are not new and can be found in different articles related to Queyranne's algorithm for SSF minimization [15, 21, 17].

4.1. Symmetric crossing submodular functions.

Consider two different sets $A, B \subseteq V$. We say that A and B are *intersecting* if $A \setminus B$, $B \setminus A$, and $A \cap B$ are all nonempty. We further say that A and B are *crossing* if they are intersecting and the set $V \setminus (A \cup B)$ is also nonempty. Consider the following submodular inequality:

$$f(A \cup B) + f(A \cap B) \le f(A) + f(B). \tag{4.1}$$

A set function $f\colon 2^V \to \mathbb{R}$ is called *fully (resp., intersecting, crossing) submodular* if inequality (4.1) is satisfied for every pair A and B in V (resp., for every pair of intersecting sets or crossing sets). Fully submodular functions are what we usually denote as submodular functions. However, from this point on we keep the adjective "fully" to avoid confusion. Note that every fully submodular function is intersecting submodular and every intersecting submodular function is crossing submodular. The function f is called *fully supermodular* if $-f$ is fully submodular and *fully modular* if it is both fully submodular and fully supermodular. We extend these definitions to their intersecting and crossing versions.

Queyranne [20] gives a very simple algorithm to find pendent pairs of symmetric fully submodular functions. Consider an ordering $(v_1, \ldots, v_n)$ of the elements of V in which the first element v_1 is selected arbitrarily and the vertex v_i is the one maximizing $f(v) - f(W_{i-1} + v)$ over $V \setminus W_{i-1}$, where W_j denotes the set $\{v_1, \ldots, v_j\}$. In other words, $(v_1, \ldots, v_n)$ satisfies

$$f(v_i) - f(W_{i-1} + v_i) \ge f(v_j) - f(W_{i-1} + v_j) \text{ for all } 2 \le i \le j \le n. \tag{4.2}$$

An ordering satisfying (4.2) is called a *maximum adjacency ordering* (this ordering has also been called a "legal order").

[4] To be precise, either V_t and V_u are subsets of Y or they are subsets of $V \setminus Y$.

LEMMA 10 (Queyranne [20]). *If f is a symmetric fully submodular function on V, and $v_1 \in V$ is an arbitrarily chosen element, the last two elements (v_{n-1}, v_n) of a maximum adjacency ordering of V starting from v_1 constitute a pendent pair. Furthermore, this ordering always exists and can be found using $O(n^2)$ oracle calls and in the same running time.*

As observed by Nagamochi and Ibaraki [15], the above result requires only symmetry and crossing submodularity, and so Lemma 10 holds also for that slightly larger class of functions. Note that we can use this to find pendent pairs whose head avoids a given element s by simply setting the first element of the ordering to be s. From this we get the following observation.

COROLLARY 11. *If f is a symmetric crossing submodular function on V, then the system (V, f) is strongly PP-admissible and we can find a pendent pair whose head avoids an element for any fusion having at least three elements in $O(|V|^2)$ time and using $O(|V|^2)$ oracle calls to f.*

4.2. Weak Rizzi functions. In this section we describe a larger class of strongly PP-admissible functions. Let V be a finite ground set and $\mathcal{Q}(V)$ be the collection of disjoint pairs of subsets of V,

$$\mathcal{Q}(V) = \{(A, B)\colon A, B \subseteq V, A \cap B = \emptyset\}. \tag{4.3}$$

A *bi-set function* on V is a real function whose domain is $\mathcal{Q}(V)$. Just like the case of set functions, we assume that bi-set functions are given through a value oracle, that is, an oracle that, given a pair of disjoint sets (A, B), returns $d(A, B)$. A bi-set function d on V is *symmetric* if for all $(A, B) \in \mathcal{Q}(V)$, $d(A, B) = d(B, A)$. Every symmetric bi-set function admits a canonical symmetric set function and vice versa. Given a symmetric bi-set function d on V, the symmetric set function $f^{(d)}$ is defined as

$$f^{(d)}(A) = d(A, V \setminus A) \text{ for all } A \subseteq V. \tag{4.4}$$

Given a (not necessarily symmetric) set function f on V, the symmetric bi-set function $d^{(f)}$ is defined as

$$d^{(f)}(A, B) = \frac{1}{2}\big(f(A) + f(B) + f(\emptyset) - f(A \cup B)\big) \text{ for all } (A, B) \in \mathcal{Q}(V). \tag{4.5}$$

PROPOSITION 12. *If f is a symmetric function on V, then $f = f^{(d^{(f)})}$.*

Proof. Let $d = d^{(f)}$ and $\hat{f} = f^{(d)}$; then for all $A \subseteq V$ we have

$$\hat{f}(A) = d(A, V \setminus A) = \frac{1}{2}\left(f(A) + f(V \setminus A) + f(\emptyset) - f(V)\right) = f(A). \qquad \square$$

Observe that even for symmetric d, the functions d and $d^{(f^{(d)})}$ can be extremely different. For example, consider the nonconstant function

$$d(A, B) = |A| + |B| \text{ for all } (A, B) \in \mathcal{Q}(V).$$

In this case $f^{(d)}$ is a constant function, and $d^{(f^{(d)})}$ is also constant.

A natural example of a set and bi-set function is the following: Let $G = (V, E, w)$ be a weighted graph with $w\colon E \to \mathbb{R}$. For all pairs (A, B) of disjoint subsets of V, let $E(A : B)$ be the set of edges with one endpoint in A and one endpoint in B. The *cut between* A *and* B is defined as $d_G(A,B) = w(E(A:B)) = \sum_{e\in E(A:B)} w(e)$, and the *cut function* of G (also called the *weighted degree function* of G) is $f_G\colon V \to \mathbb{R}$ defined as $f_G(A) = w(E(A : V \setminus A)) = \sum_{e\in E(A:V\setminus A)} w(e)$ for all $A \subseteq V$. It is easy to see that $f^{(d_G)} = f_G$ and $d^{(f_G)} = d_G$.

Rizzi [21] introduces a nice family of bi-set functions, which we describe below. We say that a symmetric bi-set function d on V is a *Rizzi bi-set function* if the following properties hold.

1. (*Consistency*) For all $A, B, C \subseteq V$ disjoint,

$$d(A,B) \le d(A,C) \text{ implies } d(A\cup C,B) \le d(A\cup B,C).$$

2. (*Monotonicity*) For all nonempty disjoint sets $A, B, C \subseteq V$,

$$d(A,B) \le d(A,B\cup C).$$

The associated *Rizzi set function* is $f = f^{(d)}$. As the following lemma shows, consistency is a natural property to consider.

PROPOSITION 13. *For every set function f, the bi-set function $d^{(f)}$ is symmetric and consistent.*

Proof. The function $d^{(f)}(A,B) = \frac{1}{2}(f(A) + f(B) + f(\emptyset) - f(A\cup B))$ is symmetric by definition. Consistency holds since

$$\begin{aligned} d^{(f)}(A\cup C,B) - d^{(f)}(A\cup B,C) &= \frac{1}{2}(f(A\cup C) + f(B) - f(A\cup B) - f(C)) \\ &= d^{(f)}(A,B) - d^{(f)}(A,C). \qquad \square \end{aligned}$$

Rizzi set functions and symmetric submodular functions are related by the following observation.

LEMMA 14 (Rizzi [21]). *If f is an intersecting submodular function on V, then the associated bi-set function*[5] *$d^{(f)}$ is a Rizzi bi-set function. Furthermore, if f is symmetric, then f is the associated Rizzi set function of $d^{(f)}$.*

Proof. Let A, B, and C be nonempty pairwise disjoint sets. Then

$$2\big(d^{(f)}(A,B\cup C) - d^{(f)}(A,B)\big) = f(B\cup C) - f(A\cup B\cup C) - f(B) + f(A\cup B). \tag{4.6}$$

Let $D = B\cup C$ and $E = A\cup B$, so that $D\cup E = A\cup B\cup C$ and $D\cap E = B$. Since A, B, and C are nonempty, D and E are intersecting. By the intersecting submodularity of f, (4.6) above is nonnegative, implying monotonicity of $d^{(f)}$. Thus $d^{(f)}$ is a Rizzi bi-set function. Furthermore, by Proposition 12, if f is symmetric, $f = f^{(d^{(f)})}$. $\square$

Rizzi [21] mistakenly states that the lemma above holds if we replace the intersecting submodularity of f by the weaker condition of crossing submodularity. However, for the symmetric crossing submodular function $f\colon V \to \mathbb{R}$ satisfying $f(\emptyset) = f(V) = 1$ and $f(X) = 0$ for all $X \notin \{\emptyset, V\}$, where $|V| \ge 3$, this does not holds. Indeed, if $\{A, B, C\}$ is a partition of V in nonempty parts, then

$$\begin{aligned} 2d^{(f)}(A,B\cup C) &= f(A) + f(B\cup C) + f(\emptyset) - f(A\cup B\cup C) = 0, \text{ and} \\ 2d^{(f)}(A,B) &= f(A) + f(B) + f(\emptyset) - f(A\cup B) = 1. \end{aligned}$$

Hence, we do not have monotonicity. One way to fix this problem is to define a weaker version of monotonicity. Consider the following property.

[5] Rizzi used the function $d(A,B) = f(A)+f(B)-f(A\cup B)$ instead, but the function d we consider is better behaved as $f = f^{(d^{(f)})}$.

1'. (*Weak monotonicity*) $d(A, B) \leq d(A, B \cup C)$ for all nonempty disjoint sets $A, B, C \subseteq V$ with $A \cup B \cup C \neq V$.

We call d a *weak Rizzi bi-set function* if it is symmetric, weak monotone, and consistent. The associated function $f^{(d)}$ is called a *weak Rizzi set function.*

LEMMA 15. *If f is a crossing submodular function on V, then the associated bi-set function $d^{(f)}$ is a weak Rizzi bi-set function. Furthermore, if f is symmetric, then f is the associated weak Rizzi set function of $d^{(f)}$.*

Proof. We need only show weak monotonicity. Indeed, for A, B, and C satisfying $A \cup B \cup C \neq V$, the right-hand side of (4.6) is nonnegative by the crossing submodularity of f. ☐

We remark here that recognizing whether a function f is a (weak) Rizzi function or not, without having access to the Rizzi bi-set function d for which $f = f^{(d)}$, is not simple. This follows since it is not always the case that $d = d^{(f^{(d)})}$. It is also easy to find Rizzi functions that are not crossing submodular (see Example 1 in section 7).

By extending the notion of fusions to bi-set functions in a natural way, it is easy to see that if d is a (weak) Rizzi bi-set function, then so are all its fusions.

Rizzi has shown that a version of Queyranne's maximum adjacency ordering allows us to find pendent pairs for Rizzi set functions. His proof, which we describe for completeness below, naturally extends to weak Rizzi set functions.

Let d be a weak Rizzi bi-set function. An ordering $(v_1, \ldots, v_n)$ of the elements of V, such that

$$d(v_i, W_{i-1}) \geq d(v_j, W_{i-1}) \text{ for all } 2 \leq i \leq j \leq n, \tag{4.7}$$

where v_1 is chosen arbitrarily and W_i denotes the set $\{v_1, \ldots, v_i\}$, is called a *maximum adjacency ordering* for d. The origin of the name comes from its interpretation for the cut function on a graph. If $d(A, B)$ represents the cut between A and B in a given graph G, then the ith vertex of a maximum adjacency ordering of d is exactly the one having a maximum number of edges going toward the previous $i - 1$ vertices. Furthermore, note that if f is a crossing submodular function on V, then the maximum adjacency orderings of f in the sense of Queyranne (4.2) coincide with those of $d^{(f)}$ in the sense of Rizzi (4.7).

LEMMA 16 (essentially in Rizzi [21]). *Let d be a weak Rizzi bi-set function on V and v_1 be an arbitrary element of V. The last two elements (v_{n-1}, v_n) of a maximum adjacency ordering of V starting from v_1 constitute a pendent pair for $f^{(d)}$. Furthermore, this ordering can be found by using $O(n^2)$ oracle calls to d and in the same running time.*

Proof. It is easy to check the claim regarding the running time since to construct the ith element of the ordering we need only find the maximum of the $n - i$ different values $\{d(x, W_{i-1})\}_{x \in V \setminus W_{i-1}}$. We show the rest by induction on the number of elements.

The lemma holds trivially for $n = 2$ since the only set separating v_1 and v_2 are singletons and the function $f^{(d)}$ is symmetric. For $n = 3$, the only sets separating v_2 and v_3 are $\{v_3\}$, $\{v_1, v_3\}$, and their complements. By definition of the ordering, $d(v_2, v_1) \geq d(v_3, v_1)$. Consistency implies that

$$f^{(d)}(\{v_1, v_3\}) = d(\{v_1, v_3\}, v_2) \geq d(\{v_1, v_2\}, v_3) = f^{(d)}(v_3).$$

Consider then $n \geq 4$, and let S be any set separating v_n and v_{n-1}. We must show that

$$d(S, V \setminus S) \geq d(v_n, V - v_n). \tag{4.8}$$

It is easy to see that $(v_{\{1,2\}}, v_3, \ldots, v_n)$ is a maximum adjacency ordering for the function $d_{1,2}$ obtained by fusing v_1 and v_2 into $v_{\{1,2\}}$. If S does not separate v_1 and v_2, then (4.8) holds since (v_{n-1}, v_n) is a pendent pair of $d_{1,2}$ by induction. So assume that S separates v_1 and v_2.

We claim that the ordering $(v_1, v_{\{2,3\}}, \ldots, v_n)$ is a maximum adjacency ordering for the function $d_{2,3}$ obtained by fusing v_2 and v_3 into $v_{\{2,3\}}$. Indeed, we need only prove that $d_{2,3}(v_{\{2,3\}}, v_1) \geq d_{2,3}(v_j, v_1)$ for all $j \geq 4$. This follows since, by hypothesis and weak monotonicity,

$$d(v_j, v_1) \leq d(v_2, v_1) \leq d(\{v_2, v_3\}, v_1).$$

If S does not separate v_2 and v_3, then (4.8) holds by induction, since (v_{n-1}, v_n) is a pendent pair for $d_{2,3}$. The only remaining possibility is the case in which S separates v_1 from v_2 and v_2 from v_3. This means that S does not separate v_1 and v_3. To conclude (4.8) it suffices to show that $(v_2, v_{\{1,3\}}, \ldots, v_n)$ is a maximum adjacency ordering for the function $d_{1,3}$ obtained by fusing v_1 and v_3 into $v_{\{1,3\}}$. Assume that this is not the case; then we must have

$$d_{1,3}(v_{\{1,3\}}, v_2) < d_{1,3}(v_j, v_2)$$

for some $j \geq 4$. Since $(v_1, \ldots, v_n)$ is a maximum adjacency ordering of d, we have $d(v_2, v_1) \geq d(v_3, v_1)$ and $d(v_3, \{v_1, v_2\}) \geq d(v_j, \{v_1, v_2\})$. By consistency we have $d(\{v_1, v_3\}, v_2) \geq d(\{v_1, v_2\}, v_3)$. Combining the inequalities above and using weak monotonicity, we get

$$\begin{aligned} d(v_3, \{v_1, v_2\}) \geq d(v_j, \{v_1, v_2\}) \geq d(v_j, v_2) &= d_{1,3}(v_j, v_2) \\ &> d_{1,3}(v_{\{1,3\}}, v_2) = d(\{v_1, v_3\}, v_2) \geq d(v_3, \{v_1, v_2\}), \end{aligned}$$

which is a contradiction. ∎

By using that the element v_1 of a maximum adjacency ordering can be chosen arbitrarily, we conclude the following result.

COROLLARY 17. *Consider the function $f = f^{(d)}$, where d is a weak Rizzi biset function on V. The system (V, f) is strongly PP-admissible, and we can find a pendent pair whose head avoids an element in any fusion having at least three elements in $O(|V|^2)$ time and using $O(|V|^2)$ oracle calls to d.*

4.3. Extensions and restrictions. Consider a set function f on V; let $S \subseteq V$ and $\overline{S} = V \setminus S$. The function obtained from f by *deleting* $\overline{S}$, also known as the *restriction* of f to S, is

$$f \setminus \overline{S} = f|_S \colon 2^S \to \mathbb{R}, \text{ where } f|_S(A) = f(A) \text{ for all } A \subseteq S. \tag{4.9}$$

We say that a restriction is *nontrivial* if the associated set S above satisfies $\emptyset \subset S \subset V$. Furthermore, any set function g such that $f = g|_V$ is called an *extension* of f.

Note that the set of solutions (in fact, the entire structure) of an arbitrary hereditary system $(V, f, \mathcal{I})$ is the same as that of the system $(W, g, \mathcal{I})$, where g is an arbitrary extension of f. Therefore, one strategy to solve a given hereditary minimization problem $(V, f, \mathcal{I})$ is to find a strongly PP-admissible extension g of f for which we can find pendent pairs avoiding a single element. It is easy to see that we need only look for extensions having only one extra element (since by fusing all the extra elements into one, we obtain another such extension).

Nagamochi and Ibaraki [15] follow a similar route and give a modification of Queyranne's pendent pair based algorithm to find all the minimal minimizers of the class of restrictions of crossing SSFs in the unconstrained setting. For that, they introduce a well-behaved type of extension.

Let f be an arbitrary set function on V and s be an element outside V. The *antirestriction* of f with extra element s is the symmetric function g on $V+s$ defined as

$$g(A) = \begin{cases} f(A) & \text{if } s \notin A, \\ f(V \setminus A) & \text{if } s \in A. \end{cases} \tag{4.10}$$

Note that g is the only extension of f by one extra element that is symmetric. Nagamochi and Ibaraki [15] and also Narayanan [17] show that f is a nontrivial restriction of a symmetric (crossing) submodular function if and only if its antirestriction g is symmetric (crossing) submodular and they use this fact to devise algorithms that work, in the unconstrained setting, on this class of functions. Furthermore, they give a nice characterization of this class of functions, which we describe below.

A set function $f\colon 2^V \to \mathbb{R}$ is called *fully posimodular* if, for all $A, B \subseteq V$,

$$f(A \setminus B) + f(B \setminus A) \leq f(A) + f(B). \tag{4.11}$$

The function f is called *intersecting posimodular* or *crossing posimodular* if inequality (4.11) is satisfied for every pair A and B of intersecting sets or crossing sets, respectively.

Nagamochi and Ibaraki show that the nontrivial restrictions of symmetric crossing submodular functions are exactly those functions that are both intersecting submodular and intersecting posimodular. This type of function appears very often: For example, the sum of a symmetric submodular function with a modular function is clearly posimodular, but it is not necessarily symmetric.

The discussion in this section shows that our methods can be used to find the minimal optimal solutions of any system $(V, f, \mathcal{I})$ provided that f is intersecting posimodular and intersecting submodular or, more generally, if f is the restriction of a weak Rizzi function. A more formal statement of this will follow Theorem 18.

5. Main results and cohereditary minimization. By specializing our methods to the families of strongly PP-admissible functions we have studied so far and to their restrictions, we obtain the following results.

THEOREM 18. *We can compute all the minimal optimal solutions of the hereditary system $(V, f, \mathcal{I})$ in time $O(|V|^3)$ and using $O(|V|^3)$ oracle calls for the following cases:*

1. *If f is symmetric crossing submodular on V, provided oracle access to f and $\mathcal{I}$.*
2. *If $f = f^{(d)}$, where d is a weak Rizzi bi-set function on V, provided oracle access to d and $\mathcal{I}$.*
3. *If f is intersecting submodular and intersecting posimodular on V, provided oracle access to f and $\mathcal{I}$.*
4. *If f is defined as*

$$f(A) = d(A, (V \setminus A) + s) \textit{ for all } A \subseteq V,$$

for some weak Rizzi bi-set function d on $V+s$, provided oracle access to d and $\mathcal{I}$.

Proof. For all results we use Algorithm FindAllMinimals and Theorem 9. The first and second results follow from Corollaries 11 and 17, respectively.

To obtain the third result, we consider the system $(V+s, g, \mathcal{I})$, where g is the antirestriction of f with extra element s. Since g is symmetric crossing submodular, we can use the first result of this theorem to find all minimal optimal solutions of $(V+s, g, \mathcal{I})$ which coincide with those of $(V, f, \mathcal{I})$.

For the last result, we use the second result to find the minimal optimal solutions of the system $(V+s, g, \mathcal{I})$, where $g = f^{(d)}$ is the weak Rizzi function on $V+s$ associated to d. Since $f = g|_V$, these solutions coincide with those of $(V, f, \mathcal{I})$. ∎

Note that Theorems 1 and 2 in the introduction follow immediately from the theorem above.

It is worth noting at this point that we can also use our methods to find all the nontrivial inclusionwise maximal minimizers of some functions constrained to *cohereditary* families, that is, families of sets closed under union. Given a family of sets $\mathcal{I}$ on V, its *dual* with respect to V is the family $\mathcal{I}^{*V} = \{A : V \setminus A \in \mathcal{I}\}$. Then, the cohereditary families are exactly the duals of hereditary families. The set function dual f^{*V} of f is defined as $f^{*V}(A) = f(V \setminus A)$ for all $A \subseteq V$.

A triple $(V, f, \mathcal{J})$ is a cohereditary system if $(V, f^{*V}, \mathcal{J}^{*V})$ is a hereditary system. A set $X \subseteq V$ is a nontrivial *maximal optimal* solution for a cohereditary system $(V, f, \mathcal{J})$ if X is an inclusionwise *maximal* set minimizing f over all the sets in $\mathcal{J} \setminus \{V\}$. Note that the maximal optimal solutions for a cohereditary system $(V, f, \mathcal{J})$ are exactly the complements of the minimal optimal solutions of the hereditary system $(V, f^{*V}, \mathcal{J}^{*V})$. This property is useful in the setting we describe next.

Consider an arbitrary set function f on V. The function obtained from f by *contracting* $\overline{S} = V \setminus S$, also known as the *contraction* of f to S, is

$$f/\overline{S} = (f \times S) \colon 2^S \to \mathbb{R}, \text{ where } (f \times S)(A) = f(A \cup \overline{S}) - f(\overline{S}) \text{ for all } A \subseteq S. \tag{5.1}$$

It is easy to see that deletion and contraction commute; that is, if S and T are disjoint subsets of V, then $(f/S) \setminus T = (f \setminus T)/S$. Any function obtained from f by deleting and contracting subsets is called a *minor* of f.

Narayanan [17] shows that every submodular function is a translation of a minor of a symmetric crossing submodular function, making the problem of finding minimizers of minors of a symmetric crossing submodular function equivalent to the corresponding one for general submodular functions. Rizzi [22] has given an example of a simple nonsymmetric fully submodular function without pendent pairs, ruling out this type of approach to minimize general submodular functions. Nevertheless, Narayanan is able to modify Queyranne's routine to find particular minimizers of *contractions* of symmetric crossing submodular functions (in the unconstrained setting).

By using the discussion in the preceding paragraphs, we can find all the maximal optimal solutions of a cohereditary system $(S, g, \mathcal{J})$ if g is a (not necessarily nontrivial) contraction of a symmetric strongly PP-admissible function f for which we can find pendent pairs whose head avoids an element in every fusion (this includes not only Narayanan's case, where g is the contraction of a symmetric crossing submodular function, but also the case where g itself is a weak Rizzi set function or a contraction of one). Indeed, if g is the contraction $f \times S$, where f is a set function on V, then its dual $g^{*S} \colon S \to \mathbb{R}$ is such that $g^{*S}(X) = g(S \setminus X) = f((S \setminus X) \cup \overline{S}) - f(\overline{S}) = f(V \setminus X) - f(\overline{S}) = f|_S(X) - f(\overline{S})$. In other words, g^{*S} is just a translation of the function $f|_S$, and so finding the maximal optimal solutions of $(S, g, \mathcal{J})$ is the same as finding the minimal optimal solutions of $(S, f|_S, \mathcal{J}^{*S})$ which we can do by Theorem 18. The next corollary summarizes our results regarding typical cohereditary systems.

COROLLARY 19. *We can compute all the maximal optimal solutions of the cohereditary system* $(V, g, \mathcal{J})$ *in time* $O(n^3)$ *and using* $O(n^3)$ *oracle calls for the following cases:*

1. *If* g *is symmetric crossing submodular on* V*, provided oracle access to* g *and* $\mathcal{J}$*.*
2. *If* $g = g^{(d)}$*, where* d *is a weak Rizzi bi-set function on* V*, provided oracle access to* d *and* $\mathcal{J}$*.*
3. *If* g^{*V} *is intersecting submodular and intersecting posimodular on* V*, provided oracle access to* g *and* $\mathcal{J}$*.*
4. *If* g *is defined as*

$$g(A) = d(A + s, V \setminus A) \text{ for all } A \subseteq V,$$

for some weak Rizzi bi-set function d *on* $V + s$*, provided oracle access to* d *and* $\mathcal{J}$*.*

Proof. The cohereditary systems $(V, g, \mathcal{J})$ for each case are exactly the duals of the hereditary systems considered in Theorem 18. ∎

6. Extreme subsets. A set $X \subseteq V$ is said to be an extreme subset of a system (V, f) if

$$f(X) < f(Y) \text{ for all } \emptyset \subset Y \subset X.$$

Every singleton $\{v\}$ with $v \in V$ is an extreme subset by definition. The following observation highlights the importance of extreme subsets for solving hereditary minimization problems.

PROPOSITION 20. *Every minimal optimal solution of the hereditary system* $(V, f, \mathcal{I})$ *is an extreme subset of* (V, f)*.*

Therefore, if we had an algorithm that reports the entire collection of extreme subsets of (V, f), and if this collection itself is not too large, we could find the minimal optimal solutions of $(V, f, \mathcal{I})$ by simply keeping the extreme subsets of (V, f) that are in the hereditary family and reporting those having minimum f-value.

Efficiently finding extreme subsets of "nice" set functions, such as the cut function of a graph or hypergraph, is a problem that has been studied by many authors. For references, see Nagamochi's article [16]. In particular, Nagamochi has recently devised a beautiful algorithm that computes all extreme subsets of (restrictions of) symmetric crossing submodular functions. This algorithm shares many similarities with Queyranne's routine; however, it is based not on finding pendent pairs but on finding a different structure called *flat pairs*. We briefly visit Nagamochi's algorithm at the end of this section.

In what follows, we show that it is possible to use a pendent pair based approach to compute all the extreme subsets of a strongly PP-admissible system, provided the collection of extreme subsets is *laminar*. A family $\mathcal{L}$ of sets is called laminar if no pair of sets in $\mathcal{L}$ is intersecting. As Example 2 in section 7 shows, the family of extreme subsets of strongly PP-admissible systems is not necessarily laminar. However, the systems studied in this paper satisfy this property, as the following lemma shows.

LEMMA 21. *Let* d *be a weak Rizzi bi-set function on* V *and* $f = f^{(d)}$ *be its associated set function. The extreme subsets of the system* (V, f) *form a laminar family.*

Proof. Let S and T be two intersecting extreme subsets of f. Consider the nonempty sets $A = V \setminus T$, $B = T \setminus S$ and $C = S \cap T$. By definition of extreme subsets, we have that $f(T \setminus S) > f(T)$ or equivalently, $d(B, A \cup C) > d(B \cup C, A)$. Using the contrapositive of consistency we deduce that $d(B, C) > d(A, C)$, i.e. $d(T \setminus S, S \cap T) > d(V \setminus T, S \cap T)$. Furthermore, by weak monotonicity, we have $d(V \setminus T, S \cap T) \geq d(S \setminus T, S \cap T)$ implying that

$$d(T \setminus S, S \cap T) > d(S \setminus T, S \cap T).$$

But, by exchanging the roles of S and T, we get the reverse inequality, which is a contradiction. ∎

The lemma above also holds for restrictions of weak Rizzi functions, including for instance, functions that are both intersecting posimodular and intersecting submodular. As every laminar family on V contains at most $2|V| - 1$ sets, it is reasonable to ask for an algorithm that returns all the extreme subsets of weak Rizzi functions in polynomial time.

More generally, suppose that (V, f) is a system whose extreme subsets form a laminar family, and $\mathcal{A}$ is an algorithm that given a hereditary family $\mathcal{I}$ on V, returns all minimal optimal solution of the system $(V, f, \mathcal{I})$. We can use $\mathcal{A}$ to compute all extreme subsets of (V, f) as follows: Let $\mathcal{I}_0 = 2^V$ and define iteratively for all $i \geq 1$,

$$\mathcal{Y}_i = \mathcal{A}(\mathcal{I}_i) \text{ (family of minimal optimal solutions of } (V, f, \mathcal{I}_i)),$$
$$\mathcal{X}_i = \bigcup_{j \leq i} \mathcal{Y}_j, \quad \mathcal{I}_{i+1} = \{A \subseteq V : \forall X \in \mathcal{X}_i, (A \cap X \neq \emptyset \Rightarrow A \subset X)\}.$$

To conclude, return $\mathcal{X}_j$, where j is the first index for which $\mathcal{Y}_j$ is empty.

Each family $\mathcal{I}_{i+1}$ defined above is hereditary and is obtained from $\mathcal{I}_i$ by *removing* the sets in $\mathcal{Y}_i$ and also all those sets that are intersecting with some set in $\mathcal{Y}_i$. In particular, the family $\mathcal{Y}_i$ consists of sets that are not inside $\mathcal{X}_{i-1}$. Note also that $\mathcal{Y}_i$ is empty only when $\mathcal{I}_i = \{\emptyset\}$. Therefore, the sequence $\{\mathcal{I}_i\}_{i \geq 0}$ is a (strictly) decreasing family that converges to $\{\emptyset\}$. That is, there is a finite index j for which:

$$2^V = \mathcal{I}_0 \supset \mathcal{I}_1 \supset \cdots \supset \mathcal{I}_j = \{\emptyset\} = \mathcal{I}_{j+1} = \cdots .$$

The index of the set $\mathcal{X}_j$ we return is exactly the index j above, meaning that our algorithm is finite. Let us check correctness.

PROPOSITION 22. *Let $\mathcal{X}$ be the family of extreme subsets of (V, f). Then for all i, $\mathcal{Y}_i \subseteq \mathcal{X}$.*

Proof. The property holds for $i = 0$. Assume by induction that it holds for a given i and let $A \in \mathcal{Y}_{i+1} \subseteq \mathcal{I}_{i+1}$. If A is not extreme, then there is a strict nonempty subset $B \subset A$ with $f(B) < f(A)$. But since $A \in \mathcal{I}_{i+1}$, we also have $B \in \mathcal{I}_{i+1}$, contradicting that A is a minimal optimal solution of $(V, f, \mathcal{I}_{i+1})$. ∎

It is easy to see that the family $\mathcal{Y}_i$ consists of the collection of extreme subsets that have minimum f-value among those not yet added to $\mathcal{X}_{i-1}$. The following proposition shows that every extreme subset is eventually added to $\mathcal{X}_i$.

PROPOSITION 23. *Let $\mathcal{X}$ be the family of extreme subsets (V, f). Then $\bigcup_{i \geq 0} \mathcal{X}_i = \mathcal{X}$.*

Proof. By the previous proposition, $\bigcup_{i \geq 0} \mathcal{X}_i \subseteq \mathcal{X}$. Suppose that there is a set $A \in \mathcal{X} \setminus \bigcup_{i \geq 0} \mathcal{X}_i$. Find the last index i such that $A \in \mathcal{I}_i$. Since $A \notin \mathcal{I}_{i+1}$ we know that there is a set $X \in \mathcal{Y}_i$ such that $A \cap X \neq \emptyset$ and $A \not\subset X$. The set A cannot contain X, since in this case, as A is extreme, $f(A) < f(X)$ and so X is not a minimal minimizer of $(V, f, \mathcal{I}_i)$. The only other possibility is that X and A are intersecting, but this cannot happen since both sets are extreme. ∎

Since in every iteration we find at least one new extreme subset, this algorithm uses at most $O(|V|)$ calls to $\mathcal{A}$. By specializing this algorithm to the families studied in this paper, we have the following results.

THEOREM 24. *We can compute all extreme subsets of the system (V, f) in time $O(|V|^4)$ and using $O(|V|^4)$ oracle calls for the following cases:*

1. *If f is symmetric crossing submodular on V, provided oracle access to f and $\mathcal{I}$.*
2. *If $f = f^{(d)}$, where d is a weak Rizzi bi-set function on V, provided oracle access to d and $\mathcal{I}$.*
3. *If f is intersecting submodular and intersecting posimodular on V, provided oracle access to f and $\mathcal{I}$.*
4. *If f is defined as*

$$f(A) = d(A, (V \setminus A) + s) \text{ for all } A \subseteq V,$$

for some weak Rizzi bi-set function d on $V + s$, provided oracle access to d and $\mathcal{I}$.

Proof. These results follows immediately by using the algorithm of Theorem 18 as a subroutine for the the algorithm described in this section. ∎

As stated at the beginning of this section, Nagamochi [16] has devised a combinatorial algorithm to find all extreme sets of crossing submodular functions (and their restrictions). In what follows we compare our results with his.

Nagamochi's algorithm is based on the following concept: We say that an unordered pair of elements $\{t, u\}$ of V is a flat pair of (V, f) if

$$f(X) \geq \min_{x \in X} f(x) \text{ for all } X \subseteq V \text{ separating } t \text{ and } u. \tag{6.1}$$

Denote a function f *FP-admissible*[6] if f and all its fusions admit flat pairs. As noted by Nagamochi, a nonsingleton extreme set A cannot separate a flat pair $\{t, u\}$. This fact alone implies that the extreme sets of FP-admissible functions form a laminar family. Even though Nagamochi does not show this, this can be derived from his results. We give a new and direct proof of this fact below.

LEMMA 25. *The family of extreme sets of FP-admissible function f is laminar.*

Proof. Suppose by contradiction that two extreme sets A and B are intersecting. Let (V', f') be the system obtained by fusing all the elements of $A \setminus B$ into a, all the elements of $B \setminus A$ into b, all the elements of $A \cap B$ into c, and, if $V \setminus (A \cup B)$ is nonempty, all the elements of this set into d. Every pair of elements in V' is separated by either A or B. Hence none of them is a flat pair. ∎

The above lemma implies that the number of extreme sets of an FP-admissible function is $O(n)$. Nagamochi gives an algorithm that outputs all the extreme sets of any FP-admissible function provided we have access to an algorithm that finds a flat pair of any fusion of f. The algorithm is similar to Queyranne's routine in the sense that at every iteration we fuse a flat pair together. A high level overview of the algorithm is given below.

Maintain a set $\mathcal{X}$ which will contain the extreme sets of f that are completely inside V_x for some $x \in V'$, where V' is the current ground set. Initialize $\mathcal{X}$ with all

[6] In [16] this class of functions is not named.

the singletons of V, and set (V', f') as (V, f). Iteratively, until the system (V', f') contains only one element, find a flat pair $\{t, u\}$ of (V', f') and fuse t and u into a single element tu. Then, test if the set V_{tu} associated to the new singleton is extreme by using the information already in $\mathcal{X}$. If V_{tu} is extreme, add it to $\mathcal{X}$. At the end of the algorithm, the family $\mathcal{X}$ contains all the extreme sets.

Nagamochi gives an efficient implementation of the above algorithm that runs in $O(nT(n))$ time, where $T(n)$ is the time needed to find a flat pair of a function over a ground set of n elements.

It is an interesting question to decide when a function is FP-admissible. Nagamochi has shown that symmetric crossing submodular functions (and also their restrictions) admit flat pairs and that we can find them as the last two elements of a so-called *minimum degree ordering* of V. He further shows that this ordering can be found in time $O(n^2)$ and using $O(n^2)$ oracle calls to f. In particular, by using this procedure as a subroutine, Nagamochi's algorithm can find all the extreme subsets of (V, f) in cubic time and using a cubic number of oracle calls. Even though this algorithm is more efficient than the pendent pair based algorithm for extreme sets we have previously presented, we would like to point out that the scopes of pendent pair based and flat pair based algorithms are different. For example, our algorithm is able to find extreme sets of weak Rizzi functions, and it is not clear whether or not there is an efficient algorithm to find flat pairs of functions in this class (we do not know if weak Rizzi functions are always FP-admissible).

Another interesting fact is that the class of strongly PP-admissible functions is incomparable with the class of FP-admissible functions. On the one hand, Example 2 in section 7 describes a strongly PP-admissible function whose extreme sets do not form a laminar family. By Lemma 25, this function is not FP-admissible. On the other hand, Example 3 in section 7 describes an FP-admissible function that does not admit pendent pairs.

7. Examples. In this section we include some relevant examples. Our first example can be found in Rizzi [21].

Example 1 (Rizzi set function that is not crossing submodular or crossing posimodular). Given a weighted graph $G = (V, E, w)$, with $w\colon E \to \mathbb{R}_+$, define the *maximum separation* between two disjoint sets of vertices A and B as

$$d(A, B) = \begin{cases} \max\limits_{ab \in E(A:B)} w(ab) & \text{if } E(A:B) \neq \emptyset, \\ 0 & \text{if } E(A:B) = \emptyset. \end{cases}$$

This function is symmetric, monotone, and consistent. Consider the complete graph in $\{a, b, c, d\}$ where all the edges have weight 1, except for ac and bd, which have weight 2. The corresponding function $f = f^{(d)}$ given by $f(A) = d(A, V \setminus A)$ for $A \subseteq V$ is not crossing submodular since

$$3 = f(\{a, c\}) + f(\{a, d\}) < f(\{a, c, d\}) + f(a) = 4.$$

The function f is not even crossing posimodular since

$$3 = f(\{a, c\}) + f(\{a, d\}) < f(c) + f(d) = 4.$$

Example 1 shows, in particular, that the class of (restrictions of) weak Rizzi functions strictly contains the class of (restrictions of) crossing submodular functions.

Example 2 (the family of extreme subsets of a strongly PP-admissible function is not necessarily laminar). Let $(V = \{a, b, c, d\}, f)$ be the symmetric function defined by

$$\begin{aligned} f(a) &= f(\{b,c,d\}) = 1, & f(\{a,b\}) &= f(\{c,d\}) = 1,\\ f(b) &= f(\{a,c,d\}) = 2, & f(\{a,c\}) &= f(\{b,d\}) = 0,\\ f(c) &= f(\{a,b,d\}) = 2, & f(\{b,c\}) &= f(\{a,d\}) = 1,\\ f(d) &= f(\{a,b,c\}) = 0, & f(\emptyset) &= f(\{a,b,c,d\}) = 0. \end{aligned}$$

Observe that the extreme subsets $\{a, c\}$ and $\{b, c\}$ cross; therefore, the extreme subsets of (V, f) do not form a laminar family. In order to show that (V, f) is strongly PP-admissible, we need the proposition below.

PROPOSITION 26. *If $|W| \le 3$ and g is symmetric, then the system (W, g) is strongly PP-admissible.*

Proof. The property holds by definition if $|W| \le 2$, so assume $W = \{a, b, c\}$. We show that there is a pendent pair whose head avoids a. Without loss of generality, assume that $g(b) \le g(c)$. By symmetry, $\min\{g(b), g(c), g(\{a, b\}), g(\{a, c\})\} = g(b)$, meaning that (c, b) is a pendent pair. $\square$

By the proposition we just proved, every nontrivial fusion of the system in Example 2 is strongly PP-admissible. To conclude that (V, f) itself is strongly PP-admissible, we show that (V, f) admits a pendent pair avoiding any of its elements. But this holds since both (b, d) and (c, a) are pendent pairs. Indeed,

$$\begin{aligned} &\min\{f(X)\colon X \cap |\{b,d\}| = 1\} \ge \min\{f(X)\colon X \subseteq V\} = f(d).\\ &\min\{f(X)\colon X \cap |\{c,a\}| = 1\} = \min\{f(a), f(\{a,b\}), f(\{a,d\}), f(\{a,b,d\})\} = f(a). \end{aligned}$$

In particular, Example 2 shows a strongly PP-admissible function that is not FP-admissible.

Example 3 (function admitting flat pairs but not pendent pairs). Let $(V = \{a, b, c\}, f)$ be the nonsymmetric set function defined by

$$\begin{aligned} f(a) &= 2, & f(\{b,c\}) &= 1,\\ f(b) &= 2, & f(\{a,c\}) &= 1,\\ f(c) &= 1, & f(\{a,b\}) &= 0,\\ f(\emptyset) &= 1, & f(\{a,b,c\}) &= 1. \end{aligned}$$

On the one hand, since $f(\{a, b\}) < f(c)$ we conclude that (a, c) and (b, c) are not pendent pairs. The pairs (c, a) and (c, b) are not pendent either since $f(c) < f(a) = f(b)$. Since $f(\{a, c\}) < f(b)$ and $f(\{b, c\}) < f(a)$, we conclude that (a, b) and (b, a) are not pendent pairs either, implying that f admits no pendent pairs.

On the other hand, the only nonsingleton sets separating a and b are $X = \{a, c\}$ and $Y = \{b, c\}$. Since $f(X) = f(Y) \ge f(c)$, we conclude that $\{a, b\}$ is a flat pair.

Acknowledgments. We would like to thank Shaddin Dughmi and Jan Vondrák for introducing us to the problem of minimizing submodular functions under cardinality constraints and for very useful discussions.

REFERENCES

[1] S. Dughmi, *Submodular Functions: Extensions, Distributions, and Algorithms: A Survey*, Ph.D. Qualifying Exam Report, Department of Computer Science, Stanford University, Stanford, CA; also available online from http://arxiv.org/abs/0912.0322, 2009.

[2] L. Fleischer and S. Iwata, *A push-relabel framework for submodular function minimization and applications to parametric optimization*, Discrete Appl. Math., 131 (2003), pp. 311–322.

[3] A. Frank, *On the edge-connectivity algorithm of Nagamochi and Ibaraki*, Laboratoire Artemis, IMAG, Université J. Fourier, Grenoble, France, 1994.

[4] G. Goel, C. Karande, P. Tripathi, and L. Wang, *Approximability of combinatorial problems with multi-agent submodular cost functions*, in Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS'09, 2009, IEEE, Washington, DC, pp. 755–764.

[5] M. X. Goemans and V. S. Ramakrishnan, *Minimizing submodular functions over families of sets*, Combinatorica, 15 (1995), pp. 499–513.

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

[7] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, 2nd ed., Springer-Verlag, New York, 1993.

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

[9] S. Iwata and K. Nagano, *Submodular function minimization under covering constraints*, in Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS'09, IEEE, Washington, DC, 2009, pp. 671–680.

[10] S. Iwata and J. B. Orlin, *A simple combinatorial algorithm for submodular function minimization*, in Proceedings of the 20th Annual ACM–SIAM Symposium on Discrete Algorithms, SODA'09, ACM, New York, SIAM, Philadelphia, 2009, pp. 1230–1237.

[11] S. Iwata, *A fully combinatorial algorithm for submodular function minimization*, J. Combin. Theory Ser. B, 84 (2002), pp. 203–212.

[12] W. Mader, *Über n-fach zusammenhängende eckenmengen in graphen*, J. Combin. Theory Ser. B, 25 (1978), pp. 74–93.

[13] H. Nagamochi and T. Ibaraki, *Computing edge-connectivity in multigraphs and capacitated graphs*, SIAM J. Discrete Math., 5 (1992), pp. 54–66.

[14] H. Nagamochi and T. Ibaraki, *A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph*, Algorithmica, 7 (1992), pp. 583–596.

[15] H. Nagamochi and T. Ibaraki, *A note on minimizing submodular functions*, Inform. Process. Lett., 67 (1998), pp. 239–244.

[16] H. Nagamochi, *Minimum degree orderings*, Algorithmica, 56 (2010), pp. 17–34.

[17] H. Narayanan, *A note on the minimization of symmetric and general submodular functions*, Discrete Appl. Math., 131 (2003), pp. 513–522.

[18] J. B. Orlin, *A faster strongly polynomial time algorithm for submodular function minimization*, Math. Program., 118 (2007), pp. 237–251.

[19] M. W. Padberg and M. R. Rao, *Odd minimum cut-sets and b-matchings*, Math. Oper. Res., 7 (1982), pp. 67–80.

[20] M. Queyranne, *Minimizing symmetric submodular functions*, Math. Program., 82 (1998), pp. 3–12.

[21] R. Rizzi, *On minimizing symmetric set functions*, Combinatorica, 20 (2000), pp. 445–450.

[22] R. Rizzi, *Excluding a simple good pair approach to directed cuts*, Graphs Combin., 17 (2001), pp. 741–744.

[23] A. Schrijver, *A combinatorial algorithm minimizing submodular functions in strongly polynomial time*, J. Combin. Theory Ser. B, 80 (2000), pp. 346–355.

[24] M. Stoer and F. Wagner, *A simple min-cut algorithm*, J. ACM, 44 (1997), pp. 585–591.

[25] Z. Svitkina and L. Fleischer, *Submodular approximation: Sampling-based algorithms and lower bounds*, in Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS'08, IEEE, Washington, DC, 2008, pp. 697–706.