

ABOVE AND BELOW: MATROIDS AND GENERALIZATIONS VIA GREEDY ALGORITHMS

AMBAREESH SHYAM SUNDAR

ABSTRACT. In this paper, we explore various axiomatizations and generalizations of matroids, motivated by the goal of classifying when certain greedy algorithms produce optimal results. We discuss the cryptomorphism of matroids in terms of the greedy algorithm, then look at greedoids, a generalization of matroids that are capable of modeling an even larger set of greedy algorithms. After introducing antimatroids and polymatroids, we discuss results regarding the optimality of greedy algorithms in greedoids.

1. GREEDY ALGORITHMS

It is well known that matroids admit various equivalent axiomatizations, known as cryptomorphisms. These cryptomorphisms can provide valuable insights into the connections between different areas of mathematics. One such cryptomorphism, due to Rado and Edmonds [1], is the following.

Definition 1.1. Let E be a finite set and $\mathcal{F} \subseteq 2^E$. We call $(E, \mathcal{F})$ an *independence system* if $\mathcal{F}$ is nonempty and downwards-closed with respect to containment (that is, if $Y \in \mathcal{F}$ and $X \subseteq Y$, then $X \in \mathcal{F}$). The sets of $\mathcal{F}$ are called *independent sets*, and a maximal independent set in $\mathcal{F}$ is called a *basis*.

Theorem 1.2. Let $(E, \mathcal{F})$ be an independence system. Then $\mathcal{F}$ is the set of independent sets of a matroid if and only if, for any cost function $c : E \rightarrow \mathbb{R}$, greedily selecting minimum-cost elements that do not violate independence produces a minimum-cost basis.

Our main objective in this section is to prove Theorem 1.2. Before we do that, however, we shall take a moment to discuss a few examples.

It turns out that many problems can be formulated as matroids by way of Theorem 1.2. As a prototypical example, let $G = (V, E)$ be a graph, and $M = M(G)$ the graphic matroid of G . Without loss of generality, assume G is connected. The independent sets of M correspond to forests in G , and choosing a cost function $c : E \rightarrow \mathbb{R}$ can be thought of as assigning a weight to each edge. The problem of finding a minimum-cost basis in M is therefore equivalent to finding a minimum spanning tree (MST) in G . Theorem 1.2 says that we can find an MST by iteratively adding to our tree the edge with the smallest weight that does not create a cycle – and this is precisely the celebrated Kruskal’s algorithm! Therefore, it turns out that to prove the correctness of Kruskal’s algorithm, we need only prove Theorem 1.2 (and of course that any graph has a graphic matroid).

Let us study yet another example that might arise in a context such as operating system development: the job scheduling problem [2]. Consider a set of jobs (processes) that must execute on a (single-core) processor. Each job J_i can be described by a tuple (t_i, d_i, p_i) : t_i is the time the job takes to execute, d_i is the deadline by which the job must be completed, and p_i is the penalty incurred if the

deadline is not met. While this problem is NP-hard (in fact, it is a generalization of the knapsack problem), it admits a natural greedy solution in the special case where all the t_i are equal to 1. Let S be the set of “accepted jobs” – that is, jobs that we guarantee will be completed before their deadline. First, sort the jobs by penalty in decreasing order. At each step, construct the set S' by adding the job J_i with the highest penalty that has not yet been considered to S . Use S' to create a job schedule by ordering the jobs in S' by deadline. If every job in S' meets its deadline, then set $S \leftarrow S'$; otherwise do not modify S .

As in the case of finding a minimum spanning tree, it is not immediately obvious that this algorithm produces an optimal job schedule. However, we can translate this problem into a matroid in a natural way. Recall the following matroid construction:

Definition 1.3. Let $\mathcal{A}$ be a sequence (A_i) , where $A_i \subseteq E$. Let G be the bipartite graph with vertex set $\mathcal{A} \cup E$, and an edge between $A \in \mathcal{A}$ and $e \in E$ if and only if $e \in A$. A *partial transversal* of $\mathcal{A}$ is a set $T \subseteq E$ such that there exists an E -saturating matching in G . The *transversal matroid* of $\mathcal{A}$ is the matroid whose independent sets are the partial transversals of $\mathcal{A}$.

Using transversal matroids, we can model our job scheduling problem as follows. Let $E = \{J_i\}$, and construct $\mathcal{A}$ as follows. Let $d_{\max}$ be the last deadline. Then for each $1 \leq i \leq d_{\max}$, let A_i be the set of jobs J_j with $d_j \geq i$ (that is, A_i is the set of jobs that can be completed without penalty at time i). A set of jobs that can each be completed by their deadline corresponds to a partial transversal of $\mathcal{A}$. Let $p_{\max}$ be the largest penalty among all jobs, and define a cost function $c : E \rightarrow \mathbb{R}$ by $c(J_i) = p_{\max} - p_i$. Since the sum of all p_i is fixed, minimizing the penalties incurred is equivalent to maximizing the sum of p_i over the jobs that are completed by their deadline – and therefore equivalent to minimizing the sum of $p_{\max} - p_i$ over all such jobs. Then Theorem 1.2 tells us that the greedy algorithm must be the optimal one to solve this problem.

Having obtained some intuition regarding the implications of Theorem 1.2, we now tackle the problem of proving it. We will use the following cryptomorphism of matroids in terms of their independent sets.

Theorem 1.4. *Let $(E, \mathcal{F})$ be an independence system. Then $\mathcal{F}$ is the set of independent sets of a matroid if and only if it satisfies the augmentation axiom: For all $A, B \in \mathcal{F}$ such that $|A| = |B| + 1$, there exists an element $a \in A \setminus B$ such that $B \cup \{a\} \in \mathcal{F}$.*

Theorem 1.5. *A matroid $(E, \mathcal{F})$ satisfies the augmentation axiom if and only if all bases of $\mathcal{F}$ have equal cardinality.*

For proofs of Theorem 1.4 and Theorem 1.5, refer to [3].

We provide two proofs of Theorem 1.2 below. The first one clearly showcases the intuition behind the result, while the second presents an elegant algebraic proof using rank functions.

First proof of Theorem 1.2. First, we prove that if $M = (E, \mathcal{F})$ is a matroid, then the greedy algorithm produces a minimum-cost basis. Let r be the size of a basis of

M , and let $e_1, \dots, e_r$ be the sequence of elements chosen by the greedy algorithm. We must have that $c(e_1) \leq c(e_2) \leq \dots \leq c(e_r)$. Assume for the sake of contradiction that the greedy algorithm fails; that is, there is a set $B' = \{e'_1, \dots, e'_r\}$ such that

$$\sum_{i=1}^r c(e'_i) < \sum_{i=1}^r c(e_i).$$

Without loss of generality, assume $c(e'_1) \leq c(e'_2) \leq \dots \leq c(e'_r)$.

Then there exists some index i for which $c(e_i) > c(e'_i)$; let j be the smallest such index. Consider the independent sets $I_1 = \{e_1, \dots, e_{j-1}\}$ and $I_2 = \{e'_1, \dots, e'_j\}$. Since $|I_2| = |I_1| + 1$, by the augmentation axiom, there exists an $e'_k \in I_2 \setminus I_1$ such that $I_1 \cup \{e'_k\}$ is an independent set. Note that $c(e'_k) \leq c(e'_j)$, and by assumption, $c(e'_j) < c(e_j)$. Then $c(e'_k) < c(e_j)$, and $I_1 \cup \{e'_k\}$ is a basis – but this is impossible, since if this were the case, the greedy algorithm would have chosen to add e'_k to the independent set instead of e_j at step j . Therefore $c(e_i) \leq c(e'_i)$ for all i , and it follows that $B = \{e_1, \dots, e_r\}$ is a minimum-cost basis.

Next, we prove that if, given an independence system and any cost function, the greedy algorithm produces a minimum-cost basis, then the independence system describes the independent sets of a matroid. Once again, we proceed by contradiction. Assume such an independence system violates the augmentation axiom. Then there exist bases $B_1, B_2 \in \mathcal{F}$ such that $|B_1| < |B_2|$. Consider the cost function

$$c(e) = \begin{cases} 1 - \varepsilon & \text{if } e \in B_2 \\ 1 & \text{if } e \in B_1 \setminus B_2 \\ 2 & \text{otherwise} \end{cases}$$

where $\varepsilon > 0$ is to be fixed later. Then the total cost of B_2 is

$$|B_2| \cdot (1 - \varepsilon) = |B_2| - \varepsilon|B_2|,$$

and the total cost of B_1 is

$$|B_1 \setminus B_2| + |B_1 \cap B_2| \cdot (1 - \varepsilon) = |B_1| - \varepsilon|B_1 \cap B_2|.$$

Note that the greedy algorithm will produce B_2 as a minimum-cost basis, since the elements of B_2 are the ones with minimum cost amongst the elements of E . If we can choose ε so that

$$|B_2| - \varepsilon|B_2| > |B_1| - \varepsilon|B_1 \cap B_2|,$$

then we can show that the basis produced by the greedy algorithm does not actually have minimum cost. To achieve this, choose any ε that satisfies

$$0 < \varepsilon < \frac{|B_2| - |B_1|}{|B_1 \cap B_2|}.$$

Then the desired inequality holds, and we arrive at a contradiction. It follows that $(E, \mathcal{F})$ must indeed be a matroid. $\square$

Second proof of Theorem 1.2. For a set $S \subseteq E$, define the *rank* $r(S)$ to be the cardinality of the largest independent set contained in S . (This coincides with the definition of rank in matroids). Similarly, define the *lower rank* $\rho(S)$ as the cardinality of the smallest independent set contained in S that is maximal with respect to subsets of S . That is,

$$\rho(S) = \min\{|S'| : S' \subseteq S, S' \in \mathcal{F}, S' \cup \{s\} \notin \mathcal{F} \text{ for all } s \in S \setminus S'\}.$$

We claim that $M = (E, \mathcal{F})$ is a matroid if and only if $\rho(S) = r(S)$ for all $S \subseteq E$. To show that $\rho(S) = r(S)$ when M is a matroid, assume I is an independent subset of S with maximum cardinality, and I' is a minimum-cardinality independent subset of S that is maximal with respect to subsets of S . Then in the restriction matroid $M|_S$, I and I' are both bases. By Theorem 1.5, $|I| = |I'|$, so $\rho(S) = r(S)$. Conversely, to show that $\rho(S) \geq r(S)$ implies that M is a matroid, take $S = E$; then $\rho(E) = r(E)$ implies that all bases of $\mathcal{F}$ have the same cardinality, and by Theorem 1.5, M is a matroid.

Let $q(M)$ be the minimum value of $\rho(S)/r(S)$ over all $S \subseteq E$. It is easy to see that $\rho(S) \leq r(S)$ always holds, so $q(M) \leq 1$. The previous paragraph then shows that $q(M) < 1$ if and only if M is not a matroid.

Given a cost function c , let $C = \max_{e \in E} c(e)$. Define a new cost function $c'(e) = C - c(e)$. Then minimizing c is equivalent to maximizing c' . Let B be the basis generated by the greedy algorithm for the maximization problem, and let B^* be a basis of maximum cost. For a set $S \subseteq E$, define $c'(S) = \sum_{e \in S} c'(e)$. We claim that

$$q(M) \leq \frac{c'(B)}{c'(B^*)}.$$

Enumerate the elements of E as $e_1, \dots, e_n$ such that $c(e_1) \leq \dots \leq c(e_n)$, and therefore $c'(e_1) \geq \dots \geq c'(e_n)$. Define $E_0 = 0$, $E_i = \{e_1, \dots, e_i\}$ for $1 \leq i \leq n$, and $B_i = B \cap E_i$ and $B_i^* = B^* \cap E_i$. Furthermore, define $d_0 = c'(e_1)$, and $d_i = c'(e_i) - c'(e_{i-1})$ for $1 \leq i \leq n$. We have that

$$\begin{aligned} c'(B) &= \sum_{i=1}^n (|B_i| - |B_{i-1}|) \cdot c'(e_i) \\ &= \sum_{i=1}^n |B_i| \cdot d_i \end{aligned}$$

by expanding the telescoping sum, and similarly

$$c'(B^*) = \sum_{i=1}^n |B_i^*| \cdot d_i.$$

Note that B_i^* is an independent set contained in E_i , so $|B_i^*| \leq r(E_i)$. Also, B_i is maximal in E_i by definition of the greedy algorithm, so $|B_i| \geq \rho(E_i)$. Then

$$|B_i^*| \rho(E_i) \leq |B_i| r(E_i) \implies q(M) \leq \frac{\rho(E_i)}{r(E_i)} \leq \frac{|B_i|}{|B_i^*|}$$

Substituting this inequality into the summations for $c'(B)$ and $c'(B^*)$ produces the desired result. $\square$

2. GREEDOIDS

A variety of interesting combinatorial and algebraic structures can be obtained as generalizations of matroids by either removing or modifying some of the axioms in some particular cryptomorphism of matroids. Recall the characterization of matroids by their independent sets from Theorem 1.4. What if we replace the requirement that $(E, \mathcal{F})$ is an independence system with the requirement that it

is an *accessible set system* ($\mathcal{F}$ is accessible if, for every nonempty $S \in \mathcal{F}$, there exists an $s \in S$ such that $S \setminus \{s\} \in \mathcal{F}$)? The resulting structure (which generalizes matroids, as any independence system is accessible) is known as a *greedoid*.

To motivate the study of greedoids, we return to the problem of finding a minimum spanning tree in a connected simple graph. We have already shown (via our proof of Theorem 1.2) that Kruskal's algorithm always produces a MST. Kruskal's algorithm is only one of two well-known MST algorithms; the other is Prim's algorithm, which takes as input a graph G , a weight function w on the edges, and a root vertex v .

PRIM'S MST($G = (V, E)$, $w : E \rightarrow \mathbb{R}_{\geq 0}$, $v \in V$):

- 1 **let** $E_{\text{MST}} \leftarrow \{\}$
- 2 **let** $V_{\text{visited}} \leftarrow \{v\}$
- 3 **while** $V_{\text{visited}} \neq V$:
- 4 **let** e be a minimum-weight vertex **in** $[V_{\text{visited}}, V_{\text{visited}}^C]$
- 5 **let** u be the endpoint of e **in** V_{visited}^C
- 6 $V_{\text{visited}} \leftarrow V_{\text{visited}} \cup \{u\}$
- 7 $E_{\text{MST}} \leftarrow E_{\text{MST}} \cup \{e\}$
- 8 **return** E_{MST}

As we can see, Prim's algorithm is a greedy algorithm, just like Kruskal's; at each step, it selects the minimum-weight acceptable edge and adds it to the MST. However, there is a key difference: whereas in Kruskal's algorithm, the set of edges at each step needed only to be acyclic, Prim's also requires that it is connected. Therefore, the set of "feasible states" of Prim's algorithm does not correspond to the set of independent sets of a matroid, since we can remove an internal edge of the tree to obtain an edge set that is not connected. This set of feasible states does, however, form a greedoid in a natural way.

Definition 2.1. Let G be a directed connected graph. An *arborescence* in G is a subgraph T of G that satisfies the following property: there exists a root $r \in V(T)$ such that, for every $v \in V(T) \setminus \{r\}$, there exists a unique directed path from r to v .

Theorem 2.2. Let G be a graph, E the edge set of G , and $\mathcal{F}$ the set of edge sets of arborescences in G rooted at a particular vertex r . Then $(E, \mathcal{F})$ is a greedoid, known as the line search greedoid of G rooted at r .

Proof. First, we show that $(E, \mathcal{F})$ is accessible. Any nonempty $S \in \mathcal{F}$ is the edge set of an arborescence, which means it has a vertex with out-degree 0. Remove this vertex and the edge s it is incident to from the arborescence to get a new arborescence. The edge set of this arborescence can be written as $S' = S - \{s\} \in \mathcal{F}$.

Next, we show that $(E, \mathcal{F})$ satisfies the augmentation axiom. Let $S, T \in \mathcal{F}$ with $|S| = |T| + 1$. Then S reaches a vertex that T does not reach. Call this vertex v .

There is a unique directed path in T from r to v . Then we may add to S the first edge in that path not contained in S to obtain another edge set in $\mathcal{F}$, as desired. $\square$

Then the set of feasible states of Prim's algorithm for a graph G (starting at vertex v) is exactly the line search greedoid of G rooted at v , where G is considered as a directed graph with two directed edges, one in each direction, for each edge in the undirected version.

One might hope that there exists a result similar to Theorem 1.2 for greedoids that would allow us to prove the correctness of Prim's algorithm. Fortunately, such a result does exist – albeit one less general than Theorem 1.2, in that it does not guarantee the correctness of the greedy algorithm for all cost functions. We will hold off on stating this result until the end of this section.

Remark 2.3. *Given that arborescences and line search matroids are defined for directed graphs, one might wonder whether the greedy algorithm solves the analogue of the MST problem for directed graphs (known as the minimum spanning arborescence problem). Unfortunately, the answer is no – we will prove this as a corollary of the aforementioned theorem. One algorithm for solving the minimum spanning arborescence problem was developed by Chu and Liu; see [4] for more details.*

To demonstrate the ubiquity of greedoids, we discuss two more particularly interesting greedoid constructions that can be considered as generalizations or analogues to matroids.

2.1. Antimatroids.

Consider again a directed graph $G = (V, E)$ with a fixed root vertex r . We can construct yet another greedoid from the arborescences in G as follows: let $V' = V \setminus \{r\}$, and let $\mathcal{F}$ be the set of $S \subseteq V'$ such that $S = V(T) \setminus \{r\}$ for some arborescence T rooted at r . Then $(V', \mathcal{F})$ is a greedoid known as the *vertex search greedoid* of G rooted at r . The line search greedoid of a graph carries strictly more information than the corresponding vertex search greedoid: the edge set of an arborescence allows us to recover the vertex set, but not the other way around. Therefore, the vertex search greedoid is, in some sense, a *more permissive* set system than the line search greedoid.

This extra permissiveness allows the vertex search greedoid to satisfy a useful property: it is closed under unions. That is, if T_1 and T_2 are arborescences rooted at r , then $V(T_1) \cup V(T_2)$ is the vertex set of some arborescence rooted at r . In fact, this condition alone characterizes a special subset of greedoids that possess a variety of interesting properties.

Definition 2.1.4. Let $(E, \mathcal{F})$ be an accessible set system that is closed under unions. Then $(E, \mathcal{F})$ is called an *antimatroid*.

Theorem 2.1.5. *Every antimatroid is a greedoid.*

Proof. $(E, \mathcal{F})$ is accessible by definition, so we just need to prove that the augmentation axiom holds. Let $A, B \in \mathcal{F}$ such that $|A| = |B| + 1$. Since $\mathcal{F}$ is accessible, there exists a chain $A_0, A_1, \dots, A_k$ of subsets in $\mathcal{F}$ such that $k = |A|$, and

$$\emptyset = A_0 \subsetneq A_1 \subsetneq \dots \subsetneq A_k = A.$$

Then for all $1 \leq i \leq k$, the following properties hold:

- $|A_0 \cup B| = |B|$.
- $|A_k \cup B| \geq |B| + 1$.
- $|A_{i-1} \cup B| \leq |A_i \cup B| \leq |A_{i-1} \cup B| + 1$.

From this, we can conclude that $|A_i \cup B|$ is nondecreasing and achieves every value between $|B|$ and $|A \cup B|$. Therefore, there exists a subset $A_j \subseteq A$ such that $|A_j \cup B| = |B| + 1$. Then $A_j \setminus B$ contains one element a , and $B \cup \{a\} \in \mathcal{F}$, as desired. $\square$

In what sense are antimatroids opposite to matroids? Both matroids and antimatroids can be characterized as special cases of greedoids in a naturally antithetical way.

Theorem 2.1.6. *A greedoid $(E, \mathcal{F})$ is a matroid if and only if it satisfies the interval property without lower bounds: if $A, B \in \mathcal{F}$ and $A \subseteq B$, for all $x \in E \setminus B$,*

$$B \cup \{x\} \in \mathcal{F} \implies A \cup \{x\} \in \mathcal{F}.$$

Proof. If $(E, \mathcal{F})$ is a matroid, $\mathcal{F}$ is an independence system, so clearly it satisfies the above property. The other direction is left as an exercise. (Hint: prove that if $B \in \mathcal{F}$ and $A \subseteq B$, then $A \in \mathcal{F}$ using induction on $|A|$.) $\square$

Theorem 2.1.7. *A greedoid $(E, \mathcal{F})$ is an antimatroid if and only if it satisfies the interval property without upper bounds: if $A, B \in \mathcal{F}$ and $A \subseteq B$, for all $x \in E \setminus B$,*

$$A \cup \{x\} \in \mathcal{F} \implies B \cup \{x\} \in \mathcal{F}$$

(note that the direction of the implication is reversed from Theorem 2.1.6).

Proof. If $\mathcal{F}$ is an antimatroid, then is closed under unions; then $A \cup \{x\} \in \mathcal{F}$ means $A \cup \{x\} \cup B = B \cup \{x\} \in \mathcal{F}$, as desired. To show the other direction, let $A, B \in \mathcal{F}$. We wish to show that $A \cup B \in \mathcal{F}$. We proceed by induction on $|A|$. If $|A| = 0$, then $A \cup B = B \in \mathcal{F}$ and we are done. Now assume the statement holds for $|A| < k$. By accessibility, choose $A' \subseteq A$ such that $A' = A \setminus \{a\}$. The inductive hypothesis gives that $A' \cup B \in \mathcal{F}$. If $a \in B$, then $A' \cup B = A \cup B \in \mathcal{F}$. Otherwise, by the interval property, $A' \cup \{a\} \in \mathcal{F}$, so $A' \cup B \cup \{a\} = A \cup B \in \mathcal{F}$, as desired. $\square$

Although we will not delve too deeply into the theory of antimatroids in this paper, we shall mention one of their especially interesting properties.

Definition 2.1.8. A *closure operator* $\text{cl} : 2^E \rightarrow 2^E$ is a function that satisfies the following properties:

- (C1) $S \subseteq \text{cl}(S)$.
- (C2) If $S \subseteq S'$, then $\text{cl}(S) \subseteq \text{cl}(S')$.
- (C3) cl is idempotent.

Definition 2.1.9. A *convex geometry* is a set E equipped with a closure operator cl that satisfies the anti-exchange axiom: if $x, y \notin \text{cl}(S)$, $x \neq y$, and $y \in \text{cl}(S \cup \{x\})$, then $x \notin \text{cl}(S \cup y)$.

Intuitively, the anti-exchange axiom encodes the properties of the *convex hull* operator. That is, $\text{cl}(S)$ can be thought of as the set of points contained in the interior of the convex hull of S ; the anti-exchange axiom says that if adding x to

the set places y in the convex hull, then adding y instead must produce a “smaller” convex hull not containing x .

There is a surprising and useful duality between convex geometries and antimatroids that is not obvious at first glance.

Theorem 2.1.10. *Let $(E, \mathcal{F})$ be a set system. Then $(E, \mathcal{F})$ is an antimatroid if and only if its dual $(E, \mathcal{F}^*)$, where $\mathcal{F}^* = \{E \setminus S : S \in \mathcal{F}\}$, is the family of closed sets of a convex geometry. The closure operator cl maps a set $T \subseteq E$ to the smallest closed set containing it.*

In some sense, this is akin to the closure operator for matroids modeling the linear span operator. For a proof of this result, see [5].

2.2. Polymatroids.

Recall that matroids can be cryptomorphically defined by their rank functions [5]:

Theorem 2.2.11. *Let E be a finite set and $r : 2^E \rightarrow \mathbb{N}$ a function. Then r is the rank function of a matroid over E if and only if, for all $X, Y \subseteq E$:*

- (R1) $r(\emptyset) = 0$.
- (R2) $X \subseteq Y \implies r(X) \leq r(Y)$.
- (R3) $r(X \cap Y) + r(X \cup Y) \leq r(X) + r(Y)$ (r is submodular).
- (R4) $r(X) \leq |X|$.

Removing axiom (R4) gives us a generalization of matroids known as polymatroids.

A polymatroid induces a greedoid in a natural way. Intuitively, the rank function should encode the idea that the corresponding greedoid is accessible: for any set $A \in \mathcal{F}$ with $k = |A|$, a chain $\emptyset = A_0 \subsetneq A_1 \subsetneq \dots \subseteq A_k = A$ should have $r(A_i) = i$.

Definition 2.2.12. Let (E, r) be a polymatroid. Then the pair $(E, \mathcal{F})$, where we define

$$\mathcal{F} = \{A \subseteq E : \text{there exists a chain } \emptyset = A_0 \subsetneq \dots \subseteq A_k = A \text{ for which } r(A_i) = i\},$$

is called the *polymatroid greedoid* of (E, r) .

Theorem 2.2.13. *The polymatroid greedoid $(E, \mathcal{F})$ of (E, r) is a greedoid.*

Proof. It is obvious from the definition that $\mathcal{F}$ is accessible, so we need only verify the augmentation axiom. The proof that the augmentation axiom is tedious, involves many applications of submodularity, and is not particularly enlightening, so we omit it; the interested reader can consult [6]. $\square$

As an example of a polymatroid, let $(E, \leq)$ be a partially ordered set. Let $\mathcal{F}$ be the family of downwards-closed sets in E (“ideals” in order theory terminology). Then $(E, \mathcal{F})$ is a polymatroid greedoid (and an antimatroid greedoid). The rank function $r : 2^E \rightarrow \mathbb{N}$ has $r(S) = |\overline{S}|$, where $\overline{S}$ denotes the downwards closure of S .

2.3. The greedy algorithm for greedoids.

We finally come to the main results of this section. Let $(E, \mathcal{F})$ be a greedoid and $c : E \rightarrow \mathbb{R}$ a cost function. Unlike the matroid case, in which there is guaranteed to exist an efficient algorithm for finding a minimum cost basis (the greedy algorithm), the analogous problem for greedoids is much more difficult.

Theorem 2.3.14. *In a greedoid $(E, \mathcal{F})$ with cost function $c : E \rightarrow \mathbb{R}$, determining whether there exists a basis with cost C is NP-hard.*

Proof. This proof is taken from [7].

We reduce the vertex cover problem, which is known to be NP-hard, to the greedoid minimum-cost basis problem. Let $G = (V, E)$ be a graph. We are tasked with determining whether a vertex cover of G exists of cardinality at most $k \in \mathbb{N}$. We construct the *vertex cover antimatroid* of G as follows. Let $D = V \cup E$. A subset $S \subseteq D$ is *covered* if, for all edges $e = uv \in S$, either $u \in S$ or $v \in S$. Let $\mathcal{F}$ be the set of covered subsets of D .

We claim that $(E, \mathcal{F})$ is an antimatroid. First, $\mathcal{F}$ is accessible because any covered subset can be reduced to the empty set by first removing all the edges, then removing all the vertices. Second, $\mathcal{F}$ is closed under unions because an uncovered edge in the union of two covered sets is impossible. This proves the claim.

By Theorem 2.1.5, $(E, \mathcal{F})$ is a greedoid. Now construct a new greedoid $(E, \mathcal{F}')$, where

$$\mathcal{F}' = \{S \in \mathcal{F} : |S| \leq e(G) + k\}.$$

If there exists a set $S \in \mathcal{F}'$ with $e(G)$ edges and at most k vertices, then G has a vertex cover of cardinality at most k . To ensure this condition, set $c(e) = -1$ for every $e \in E$, and $c(v) = 0$ for every $v \in V$. Then $\mathcal{F}'$ has a basis with cost $C = -k$ if and only if G has a vertex cover of cardinality k . This completes the reduction. $\square$

Next, we state a generalization of Theorem 1.2 characterizing exactly when the greedy algorithm does succeed for greedoids.

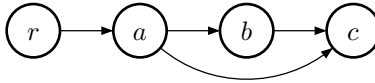
Theorem 2.3.15. *Let $(E, \mathcal{F})$ be a greedoid and $c : E \rightarrow \mathbb{R}$ a cost function. The greedy algorithm finds a minimum-cost basis if and only if $(E, \mathcal{F})$ satisfies the strong exchange property: for all $A, B \in \mathcal{F}$ where $A \subseteq B$ and B is a basis, for all $x \in E \setminus B$ with $A \cup \{x\} \in \mathcal{F}$, there exists a $y \in B \setminus A$ for which $A \cup \{y\} \in \mathcal{F}$ and $(B \cup \{x\}) \setminus \{y\} \in \mathcal{F}$.*

Proof. Consult [7]. $\square$

As a corollary, we show that an extension of Prim's algorithm to directed graphs fails.

Corollary 2.3.16. *The greedy algorithm does not solve the maximum rooted arborescence problem.*

Proof. We produce a graph whose line search greedoid does not satisfy the strong exchange property. Consider the graph



Let $(E, \mathcal{F})$ be its line search greedoid. Take $A = \{ra\}$ and $B = \{ra, ab, bc\}$. Then $A, B \in \mathcal{F}$, and B is a basis. There is only one edge $x = ac \in E \setminus B$. If we choose $y = ab$, then $A \cup \{y\} = \{ra, ab\}$ is in $\mathcal{F}$, but $(B \cup \{x\}) \setminus \{y\} = \{ra, ac, bc\}$ is not an arborescence, since it contains no directed path from r to b . Otherwise, if

we choose $y = bc$, then $A \cup \{y\} = \{ra, bc\}$ is not an arborescence because it is disconnected. Therefore this graph does not satisfy the strong exchange property, and as a result, there is some set of edge weights for which the greedy algorithm fails. Indeed, consider the weights $c(ra) = 0$, $c(ab) = 1000$, $c(bc) = 1$, $c(ac) = 2$. The greedy algorithm chooses $\{ra, ac, ab\}$ for a total cost of 1002, whereas the true minimum spanning arborescence is $\{ra, ab, bc\}$ with cost 1001. $\square$

3. CONCLUSION

Due to space constraints, we were only able to provide a fairly brief introduction to greedoids. There are many connections to various areas of mathematics that we were unable to explore in this paper; some interesting topics for further reading include:

- the greedoid polynomial;
- algorithms for determining whether a set is independent in a matroid/feasible in a greedoid;
- chip-firing games and antimatroids;
- polymatroids, their characterization in terms of greedy algorithms, and applications in combinatorial optimization.

Information about many of these topics can be found in [5] and [7], and the author strongly recommends these books, as they were invaluable resources in the preparation of this paper.

REFERENCES

1. Edmonds, J.: Matroids and the greedy algorithm. *Mathematical programming*. 1, 127–136 (1971)
2. Lawler, E.L.: *Combinatorial optimization: networks and matroids*. Courier Corporation (2001)
3. Oxley, J.G.: *Matroid theory*. Oxford University Press, USA (2006)
4. Chu, Y.-J.: On the shortest arborescence of a directed graph. *Scientia Sinica*. 14, 1396–1400 (1965)
5. *Matroid Applications*. Cambridge University Press (1992)
6. Korte, B., Lovász, L.: Polymatroid greedoids. *Journal of Combinatorial Theory, Series B*. 38, 41–72 (1985). [https://doi.org/https://doi.org/10.1016/0095-8956\(85\)90091-7](https://doi.org/https://doi.org/10.1016/0095-8956(85)90091-7)
7. Korte, B., Vygen, J.: *Combinatorial optimization: theory and algorithms*. Springer (2008)
8. Kun, J.: *When Greedy Algorithms are Perfect: the Matroid*, (2014)