

Approximation Algorithms for Matroidal Prerequisite Systems

Robert P. Streit 

Dept. of Electrical & Computer Engineering, The University of Texas at Austin, TX, USA

Vijay K. Garg 

Dept. of Electrical & Computer Engineering, The University of Texas at Austin, TX, USA

Abstract

Optimal selections in a decision process are often constrained by prerequisites. However, such prerequisites can encode functional rather than literal dependencies, so a required dependency may be supplied by one or several interacting alternatives. We introduce *matroidal prerequisite systems* (MPS), a combinatorial constraint structure where a poset specifies prerequisites while a matroid determines when those prerequisites have been satisfied by its span. This creates an order-sensitive notion of feasibility over words, where feasible words are associated with independent sets, while dependencies may be fulfilled through substitutable functionality.

Our main contribution is approximation algorithms for nonnegative additive maximization and monotone submodular maximization over the feasible words of an MPS. The guarantees are determined by two structural parameters: the maximum matroid rank Δ of a principal ideal in the poset and the maximum matroid connectivity $\lambda_{\max}$. These measure the distance an MPS is from encoding a matroid or a poset antimatroid, respectively, both of which are generalized by an MPS. For additive maximization, we obtain deterministic Δ - and $(1 + \lambda_{\max})$ -approximation algorithms. By extending these techniques, we obtain efficient deterministic $(2 + \lambda_{\max})$ -approximation and randomized $(\Delta^2 \cdot (1 - 1/e - \delta)^{-1})$ -approximation algorithms for all $\delta > 0$ for submodular maximization. The algorithm design and analysis use the theory of polymatroid greedoids, via a cryptomorphism we prove between an MPS and a *strong polymatroid greedoid*. Finally, a reduction from densest k -subgraph shows it is not possible to efficiently compute a $\min\{\Delta, \lambda_{\max}\}^{o(1)}$ -approximation to nonnegative additive maximization over the feasible words of an MPS under the Gap Exponential Time Hypothesis. Thus, an MPS provides a tractable, but provably nontrivial, framework for combinatorial optimization with interacting prerequisites, independence, and substitution.

2012 ACM Subject Classification Mathematics of computing $\rightarrow$ Combinatorial optimization; Mathematics of computing $\rightarrow$ Matroids and greedoids; Theory of computation $\rightarrow$ Approximation algorithms analysis

Keywords and phrases matroids, posets, polymatroid greedoids, submodular maximization

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2026.23

Category APPROX

Related Version *Full Version*: <https://arxiv.org/abs/2607.08151>

Funding *Vijay K. Garg*: Partially supported by the Cullen Trust for Higher Education Endowed Professorship.

1 Introduction

In a discrete decision process, some decisions are not meaningful, feasible, or useful until other functionality is already present. This is perhaps most obvious in task scheduling problems, where dependencies impose constraints on what can and cannot be completed. However, such dependencies are rarely literal. Some project task may require not an earlier task itself, but rather some interface or information that earlier tasks are meant to provide. These required



© Robert P. Streit and Vijay K. Garg;

licensed under Creative Commons License CC-BY 4.0

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2026).

Editors: Mohit Singh and Tom Gur; Article No. 23; pp. 23:1–23:24



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

functionalities may be satisfied by alternative selections of one or many interacting decisions. Yet, existing prerequisite models (e.g. [38, 19, 41, 31, 20]) typically require prerequisites to be selected explicitly; i.e. if y is a prerequisite of z , then z can only be chosen after y . So, there is no awareness of the overlap or substitutable nature of the decision functionalities.

Matroids encode such a substitution structure. A *matroid* $\mathcal{M}$ is a nonempty set family, whose members are called *independent*, such that every subset of an independent set is independent and for all independent X and Y with $|Y| > |X|$ there exists $z \in Y \setminus X$ such that $X + z$ is independent. The *rank* $r_{\mathcal{M}}(X)$ of an arbitrary set X is the size of a maximal independent set it contains, while the *span* $\sigma_{\mathcal{M}}(X)$ is the union of X with the elements which do not increase its rank. Now, if feasible sequences of decisions $x_1 \dots x_\ell$ are always such that $\{x_1, \dots, x_\ell\}$ is independent, then the span $\sigma_{\mathcal{M}}(\{x_1, \dots, x_\ell\})$ can encode the functionalities covered by $x_1 \dots x_\ell$. So, two decisions $y, z \notin \sigma_{\mathcal{M}}(\{x_1, \dots, x_\ell\})$ which are not made redundant by these functionalities (in the sense they are spanned by $\{x_1 \dots x_\ell\}$) are $\{x_1, \dots, x_\ell\}$ -*substitutable* if and only if $\sigma_{\mathcal{M}}(\{x_1, \dots, x_\ell, y\}) = \sigma_{\mathcal{M}}(\{x_1, \dots, x_\ell, z\})$. This is, given what is already covered by $x_1 \dots x_\ell$, any new functionality added by y or z leads to equivalent coverage. Matroids hold a central place in discrete optimization because their structure models many problems and gives algorithms for many settings (see [40, 10, 35, 5, 14, 45]).

In this work, we study the optimization problems that arise when prerequisite and matroidal substitution structure are forced to interact. Our reasons are both modeling based and algorithmic. Many applications lead to prerequisite structure defined via substitutable functionalities rather than irreplaceable objects, and we would like to capture this. We will also see the resulting constraint systems form a structured class weaker than matroids that still admits approximation algorithms for additive and submodular maximization. To that end, we propose the *matroidal prerequisite system (MPS)*, which is a simple language (words without repeating letters) over a finite alphabet Σ defined by an interaction between a matroid $(\Sigma, \mathcal{M})$ and poset $(\Sigma, \rightsquigarrow)$. The words $y_1 \dots y_\ell$ in the language, denoted by $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$, are such that their letters always form independent sets, and each y_i is such that all prerequisites $x \rightsquigarrow y_i$ are either spanned by the prefix $\{y_1, \dots, y_{i-1}\}$ in the matroid or x and y_i are $\{y_1, \dots, y_{i-1}\}$ -substitutable. This, along with a compatibility condition stating that letters spanned by feasible words must be such that their prerequisites are also spanned, defines an MPS. The upshot is the poset and matroid retain their usual roles, but no longer act independently. Specifically, the poset specifies which decisions require earlier prerequisite coverage, and the matroid determines when that prerequisite coverage has been supplied via its span while continuing to enforce nonredundancy via independence. An MPS generalizes matroids, as $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ is the permutations of independent sets if the ordering relation is empty, as well as *poset antimatroids* (the language of linear extensions of a poset) as $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ is the linear extensions of $(\Sigma, \rightsquigarrow)$ if $\mathcal{M}$ is free (all sets are independent).

Our main contributions are approximation algorithms for maximization over the words of an MPS of nonnegative additive and monotone submodular functions. Our guarantees are given in terms of structural parameters measuring the distance of an MPS from being a matroid or poset antimatroid. First is the largest matroid rank of a principal ideal,

$$\Delta \triangleq \max_{z \in \Sigma} r_{\mathcal{M}}(\downarrow z), \quad \text{where } \downarrow z \triangleq \{y \in \Sigma \mid y \rightsquigarrow z\} \cup \{z\}. \quad (1)$$

If the MPS is a matroid (i.e. “ $\rightsquigarrow$ ” is empty) then $\Delta = 1$, and as the prerequisites become more complex Δ will increase. The second parameter is defined by the largest value taken by the (Tutte) connectivity function λ (see [47, 48, 37]) of the matroid,

$$\lambda_{\max} \triangleq \max_{X \subseteq \Sigma} \lambda(X), \quad \text{where } \lambda(X) \triangleq r_{\mathcal{M}}(X) + r_{\mathcal{M}}(\Sigma \setminus X) - r_{\mathcal{M}}(\Sigma). \quad (2)$$

The connectivity $\lambda(X)$ computes how much the complement $\Sigma \setminus X$ can cover X in its span, in terms of rank. Equivalently, $\lambda(X)$ measures how much of X can be substituted with its complement. So, $\lambda_{\max}$ measures how complex the substitutes structure is. Note $\lambda_{\max} \geq 0$. For free matroids where every set is independent, we see $\lambda_{\max} = 0$. So, $\lambda_{\max} = 0$ when the MPS defines a poset antimatroid and increases as the substitutes structure becomes more complex. This leads to our first main result.

► **Main Result 1** (Theorems 12, 13, 14, and 15). *Let $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ be a matroidal prerequisite system. Assume we are given a matroid rank oracle.*

1. *There exists an efficient deterministic $\min\{\Delta, 1 + \lambda_{\max}\}$ -approximation algorithm for nonnegative additive maximization over the feasible words of $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$,*
2. *There exists an efficient deterministic $(2 + \lambda_{\max})$ -approximation and, for every $\delta > 0$, an efficient randomized $(\Delta^2 \cdot (1 - 1/e - \delta)^{-1})$ -approximation algorithm for monotone submodular maximization over the feasible words of $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$.*

Observe, our methods achieve exact solutions for additive maximization whenever the MPS is a matroid or poset antimatroid. These are the two “easy” cases, and the approximation smoothly degrades as we drift towards a more complex MPS structure (as Δ and $\lambda_{\max}$ will increase). For submodular maximization we achieve essentially the best possible approximation [34] when the MPS is a matroid. For poset antimatroids, the problem becomes trivialized as the entirety of Σ can always be permuted into a linear extension, so outputting this permutation always leads to an exact solution even for monotone submodular objectives. But, our guarantee would give a 2-approximation. Whether it is possible to improve this to smoothly interpolate between this scenario and more complex MPS structures (like in the additive case) is something we leave to future work. Finally, we note that the expected $(\Delta^2 \cdot (1 - 1/e - \delta)^{-1})$ -approximation guarantee comes from using the continuous greedy algorithm [5] as a black-box subroutine. We present our methods with this chosen subroutine because it is the most recognizable to the widest audience, but one could replace this with any of [45, 14, 4, 7] to obtain various improvements.

Our algorithmic techniques make use of *strong polymatroid greedoids* recently introduced in [44]. Specifically, a polymatroid greedoid is a greedoid whose feasibility structure can be encoded by a polymatroid rank function called a *representation*. Strong polymatroid greedoids are a subclass admitting certain closure operators and a canonical representation called *the greatest representation*. We prove a cryptomorphism in Theorem 9 showing a greedoid is a strong polymatroid greedoid if and only if it can be encoded as an MPS. This allows us to use the greatest representation in our algorithms after verifying this function can be evaluated efficiently using a matroid rank oracle. In particular, Section 5 presents algorithms based off of solving a relaxation over the base polytope of the greatest representation, while Section 6 uses the greatest representation to create modified greedy algorithms. The correctness will follow from Lemma 11, which discusses how to iteratively “fill-in” sets of letters contained in feasible words to converge to a full feasible word in a strong polymatroid greedoid.

To complement the algorithmic results, we also investigate the hardness of nonnegative additive maximization. We give an approximation-preserving reduction from Densest k -Subgraph. By [29], this shows our algorithmic guarantees are meaningful in that exponential improvements are not possible under the Gap Exponential Time Hypothesis (see [30, 9, 6]).

► **Main Result 2.** *Under the Gap Exponential Time Hypothesis, there is no $\min\{\Delta, \lambda_{\max}\}^{o(1)}$ -approximation algorithm for additive maximization over a matroidal prerequisite system.*

Our reduction uses uniform matroids, so we note that this hardness applies even to an MPS defined with a uniform matroid. While Main Result 2 shows constant or logarithmic-approximation algorithms in these parameters are unlikely, the result doesn’t comment on sublinear approximations in their entirety. We leave the resolution of this gap to future work.

Overview. We begin by surveying related work and making comparisons in Section 2. Then, in Section 3 we overview some background. Section 4 will introduce the MPS. Section 4.1 examines the greedoid theory of an MPS, and proves a cryptomorphism between the class of MPS and strong polymatroid greedoids from [44] in Theorem 9. Section 4.2 shows that the greatest representation of a MPS can be evaluated efficiently via a matroid rank query, while Section 4.3 proves that every sequence of letters such that the marginal return of each letter against its prefix in the greatest representation is greater than zero can be “filled” into a feasible word by repeated letter insertions.

In Section 5 we design approximation algorithms with guarantees varying in Δ . Section 5.1 gives a method for additive maximization which is based on solving a relaxation defined by the greatest representation and projecting the solution onto a feasible word using Lemma 11. This is Algorithm 1, which Theorem 12 shows is a Δ -approximation algorithm. Then, in Section 5.2 we obtain a method for monotone submodular maximization by applying the continuous greedy algorithm [5] to the *natural matroid* construction [16, 3] of the greatest representation, and using the solution to create an additive underestimation which is fed into Algorithm 3. The analysis in Theorem 13 shows this process leads to an expected approximation at most a Δ^2 factor worse than that of [5] for submodular maximization over matroids. In Section 6, we design approximation algorithms with guarantees varying in $\lambda_{\max}$. Section 6.1 begins by examining algorithms for additive objectives, and presents a modified greedy algorithm (Algorithm 3) using the greatest representation to adjust the perceived value of available letters in each round, and the closure operator of strong polymatroid greedoids to determine where to insert selected letters into a working solution. Theorem 14 shows the resulting solution is a $(1 + \lambda_{\max})$ -approximation. Then, Section 6.2 extends the technique to achieve a $(2 + \lambda_{\max})$ -approximation for submodular objective functions. Finally, we conclude by proving Main Result 2 in Section 7.

2 Related Work

Our MPS is in many ways a relaxation of matroids for designing approximation algorithms in more general settings. Other approaches have been made in this spirit. See for example p -systems [35], k -intersection [21, 27], k -extendible [32], and k -exchange [13] systems. All come with approximate guarantees decaying in the “distance” from a matroid, like the Δ parameter used in Main Result 1. But, a key difference is our feasibility model is order-sensitive. This is to say, where previous works maintain a subset-closed set family, our approach injects a notion of prerequisites using a poset that destroys such structure. This is why we interpret our MPS as a greedoid [25], as this theory accommodates order-sensitive feasibility.

On that note, there do exist works mixing prerequisite structure with other combinatorial constraint systems. A notable example is precedence constrained knapsack variants [41, 19, 20] and maximum closure [38, 31] problems. However, the prerequisite constraints of these problems require a “complete” closure, in the sense that an element can only be selected once *all* of its prerequisites are present. In contrast, we use a matroid to encode notion of substitution or redundancy. Along these lines, AND/OR scheduling problems [33] use Boolean formulae to allow for alternatives through disjunction. However, in [33] the formulae are explicit inputs, while our delegation to the matroid induces a stateful and implicit notion of substitutability which varies over the decision process.

Our work uses greedoids, a structure introduced by Korte and Lovász to generalize matroids by removing the definitional dependence on subset-closed set families [22, 25, 2]. For linear optimization, [23, 15] give a necessary and sufficient called *strong exchange* for

the correctness of the greedy algorithm in greedoids, which was later generalized to the idea of a *matroid embedding* in [17]. In Section 4 we will see that the greedy algorithm fails to solve additive maximization on an MPS, and so these results do not apply to our setting. Without strong exchange, there do exist certain characterizations on the types of objective functions which the greedy algorithm can correctly solve [25, 46, 12], but they are sensitive to the greedoid structure and quite restrictive. Polymatroid greedoids are introduced in [24], and recently studied in [44]. This class is defined by an interaction between a polymatroid and a greedoid which allows for the use of submodular analysis (which is typically not fully available in the greedoid setting, unlike matroids). The latter work [44] introduces *strong polymatroid greedoids*, which give special properties we use for the design and analysis of our algorithms after showing an MPS is cryptomorphically equivalent to such a greedoid.

3 Preliminaries

For any set X , let $X!$ give its permutations, and $\chi_X \in \{0, 1\}^\Sigma$ be its characteristic vector. We often iterate over words written like $y_1 \dots y_{i-1}$, so we are implicitly writing $y_1 \dots y_0$ when $i = 1$. Let $y_1 \dots y_0 = \epsilon$, where ϵ is the empty word, and $\{y_1, \dots, y_0\} = \emptyset$. Finally, for $v \in \mathbb{R}^\Sigma$, the *support* $\text{supp}(v) \triangleq \{y \in \Sigma \mid v(y) \neq 0\}$ is the set of labels for the nonzero coordinates in v .

Submodularity. An *additive function* is a set function defined by assigning each letter a weight via $w : \Sigma \rightarrow \mathbb{R}$ and mapping sets to the sum of the weights of letters they contain, like $X \mapsto \sum_{y \in X} w(y)$. A *submodular function* $f : 2^\Sigma \rightarrow \mathbb{R}$ is a set function satisfying the *law of diminishing returns* defined as $X \subseteq Y$ implies $(f/X)(z) \geq (f/Y)(z)$ for all $z \in \Sigma$. where $(f/X)(z) \triangleq f(X + z) - f(X)$ is the evaluation of the *contraction of f by X* at the letter z . We adopt the convention that $(f/X)(z) = 0$ whenever $z \in X$. Note, if $X = \{y_1, \dots, y_\ell\}$, then $f(X) = \sum_{i=1}^\ell (f/\{y_1, \dots, y_{i-1}\})(y_i)$. And so, submodular functions generalize additive functions. Nonnegative additive and monotone (i.e. $X \subseteq Y$ implies $f(X) \leq f(Y)$) submodular functions are the objective functions we consider. We will assume that f is *normalized*, i.e. $f(\emptyset) = 0$, to remove any translation effects from our approximation ratios. This is a common maneuver, see [35, 5, 14] for example.

A *polymatroid rank function* $\rho : 2^\Sigma \rightarrow \mathbb{Z}$ is a monotone, normalized, submodular function. Though integrality is not strictly required, the polymatroid rank functions used in this work will have integral codomains. A matroid rank function $r_{\mathcal{M}}$ is simply a *subcardinal* polymatroid rank function, i.e. $r_{\mathcal{M}}(X) \leq |X|$ always holds. In this way, polymatroids generalize matroids. They can also be viewed as a multiset generalization of matroids as well. These perspectives lead to a few different ways of constructing matroids from polymatroids, which we will see in Section 4.1 and Section 5.2. The *base polytope* of a polymatroid is the set of nonnegative vectors in $\mathbb{R}^\Sigma$ saturating the total rank and satisfying a collection of rank inequalities over all subsets like $\{v \in \mathbb{R}_+^\Sigma \mid \|v\|_1 = \rho(\Sigma) \text{ and } (\forall X \subseteq \Sigma) \sum_{y \in X} v(y) \leq \rho(X)\}$. Famously, when introducing polymatroids Edmonds gave a greedy algorithm for solving linear maximization [11]. For a solution over the base polytope, this is sorting the coordinates by weight and iteratively increasing each coordinate by its marginal return against its predecessors in the sorted order (see [11] for details). Finally, the *span* operator $\sigma_\rho : 2^\Sigma \rightarrow 2^\Sigma$ is defined as $\sigma_\rho(X) \triangleq \{y \in \Sigma \mid \rho(X + y) = \rho(X)\}$, i.e. those elements for which the marginal return follows $(\rho/X)(y) = 0$. The span is a closure operator, that is it is *monotone* (i.e. $X \subseteq Y$ implies $\sigma_\rho(X) \subseteq \sigma_\rho(Y)$), *idempotent* (i.e. $\sigma_\rho(\sigma_\rho(X)) = \sigma_\rho(X)$), and *extensive* (i.e. $X \subseteq \sigma_\rho(X)$). We note that the ρ -closed sets are closed under intersection, and that this is a necessary and sufficient description of any closure operator. And so, those sets $X \subseteq \Sigma$

such that $X = \sigma_\rho(X)$ are called ρ -closed. Notice, $\sigma_{r_M} = \sigma_M$ for a matroid M . A defining feature is the *Steinitz-Maclane exchange property*, which is $y, z \notin \sigma_M(X)$ and $y \in \sigma_M(X + z)$ implies $z \in \sigma_M(X + y)$. This is the reason why we may use matroid span operators to encode substitutability in constraint structures, as the Steinitz-Maclane exchange property ensures that z “capturing” y in the increased span $\sigma_M(X + z)$ is symmetric. Finally, a *loop* $y \in \Sigma$ is an element such that $\rho(y) = 0$.

Lattices and Order. Let $\mathcal{P} = (\mathcal{X}, \sqsubseteq)$ be a poset with reflexive closure “ $\sqsubseteq$ ”. A set $I \subseteq \mathcal{P}$ is an (order) *ideal* if $\mathbf{z} \in I$ and $\mathbf{y} \sqsubseteq \mathbf{z}$ implies $\mathbf{y} \in I$. The ideal generated by an arbitrary set S is $\downarrow S \triangleq \{\mathbf{y} \in \mathcal{X} \mid (\exists \mathbf{z} \in S) \mathbf{y} \sqsubseteq \mathbf{z}\}$. A *principal ideal* $\downarrow \mathbf{y}$ is generated by a single poset element. For $\mathbf{x}, \mathbf{y} \in \mathcal{P}$ the *meet* $\mathbf{x} \sqcap \mathbf{y}$ is the binary infimum while the *join* $\mathbf{x} \sqcup \mathbf{y}$ is the binary supremum whenever unique such elements exist. A poset is a *lattice* when a meet and join is defined for every pair. The *covering relation*, which we denote by “ $\prec$ ” for any poset, is such that $\mathbf{x} \prec \mathbf{y}$ if and only if $\mathbf{x} \sqsubseteq \mathbf{y}$ and there is no $\mathbf{z}$ with $\mathbf{x} \sqsubseteq \mathbf{z} \sqsubseteq \mathbf{y}$. One visualizes posets via *Hasse diagrams*, consisting of nodes corresponding to the elements of the ground set $\mathcal{X}$ and line segments such that there is a line directed upwards from $\mathbf{x}$ to $\mathbf{y}$ if and only if $\mathbf{x} \prec \mathbf{y}$. A poset $\mathcal{P}$ is *graded* whenever there exists a function $g : \mathcal{P} \rightarrow \mathbb{Z}_+$ such that $\mathbf{x} \prec \mathbf{y}$ implies $g(\mathbf{x}) = g(\mathbf{y}) - 1$. A grade taking value zero on bottom elements is the *height function*. Finally, a lattice is *semimodular* if and only if $\mathbf{x} \succ \mathbf{x} \sqcap \mathbf{y}$ implies $\mathbf{y} \prec \mathbf{x} \sqcup \mathbf{y}$. We note semimodular lattices are always graded.

Greedoids. Let Σ^* be all finite sequences over Σ . Each $\alpha \in \Sigma^*$ is a *word* $\alpha = y_1 \dots y_k$ composed of *letters* from Σ . A *language* $\Lambda \subseteq \Sigma^*$ is a set of words. We only examine *simple* languages which do not contain words with repeated letters. To model problems of interest using languages, one can let letters symbolize decisions which can be made over a process, while words give the *feasible* arrangements of decisions. The *support* $\tilde{\alpha}$ is the set of letters forming α , and we refer to the length of a word via $|\alpha|$. Longest words in Λ are called *basic*. Finally, a *loop* $y \in \Sigma$ is a letter such that $y \notin \tilde{\alpha}$ for all $\alpha \in \Lambda$, i.e. a loop does not appear in any word. A *greedoid* can be defined as a language generalization of a matroid.

► **Definition 1 (Greedoid).** A greedoid Λ over Σ is a nonempty simple language satisfying,

1. (Hereditary) $\alpha\beta \in \Lambda$ implies the prefix α is feasible, i.e. $\alpha \in \Lambda$,
2. (Exchange) For all $\alpha, \beta \in \Lambda$, if $|\alpha| > |\beta|$ then there exists $x \in \tilde{\alpha}$ such that $\beta x \in \Lambda$.

We give an example relevant to our MPS construction in Section 4, and point the reader towards [25, 2] for more examples. Non-emptiness and the hereditary axiom together imply $\epsilon \in \Lambda$, and exchange makes all basic words equal in length. A prominent class is *interval greedoids*, which possess a stronger variant of exchange: call a simple $\alpha' \in \Sigma^*$ a *subword* of $\alpha \in \Lambda$ if and only if $\tilde{\alpha}' \subseteq \tilde{\alpha}$ and the letters of α' occur in an order agreeing with α (i.e. x appears before y in α' implies x appears before y in α).

► **Definition 2 (Interval Property).** Let Λ be a greedoid. Λ has the interval property if for all $\alpha, \beta \in \Lambda$ with $|\beta| > |\alpha|$, there exists a subword β' of β of length $|\beta'| \geq |\beta| - |\alpha|$ with $\alpha\beta' \in \Lambda$.

The contraction minor Λ/α , where $\alpha \in \Lambda$, is the language of simple words which can be concatenated onto α while maintaining feasibility in Λ , i.e. $\Lambda/\alpha \triangleq \{\beta \in \Sigma^* \mid \alpha\beta \in \Lambda\}$. Λ/α is a greedoid, and we call the single length words $y \in \Lambda/\alpha$ *continuations* since they are exactly $y \in \Sigma$ with $\alpha y \in \Lambda$. Define the equivalence relation $\alpha \sim \beta$ by $\Lambda/\alpha = \Lambda/\beta$. A *flat* is an equivalence class of $\Lambda/\sim$, and the class associated with α is $[\alpha]$. One can order the flats by a kind of reachability: let $[\alpha] \sqsubseteq [\beta]$ if and only if there exists $\gamma \in \Lambda/\alpha$ where $\alpha\gamma \in \Lambda$.

and $\alpha\gamma \sim \beta$. For interval greedoids, the resulting poset $(\Lambda/\sim, \sqsubseteq)$ is a semimodular lattice. Finally, we define the *flat support* as the union of supports of words in a flat. This is given by $\kappa : \Lambda/\sim \rightarrow 2^\Sigma$ defined as $\kappa[\alpha] \triangleq \bigcup_{\beta \sim \alpha} \tilde{\beta}$. For interval greedoids κ is invertible, so ordering the flat supports by containment gives a lattice isomorphic to $(\Lambda/\sim, \sqsubseteq)$.

A *polymatroid greedoid*, introduced in [24], is such that there exists a polymatroid rank function $\rho : 2^\Sigma \rightarrow \mathbb{Z}$ satisfying $\Lambda = \{x_1 \dots x_k \in \Sigma^* \mid (\forall i \in \{1, \dots, k\}) \rho(\{x_1, \dots, x_i\}) = i\}$. Examples include matroids, poset antimatroids, and certain ordered geometries; see Ch. 7 of [25]. Following [44], in this context we call ρ a *representation* as it encodes how one builds feasible words in Λ . This is, for all $\alpha \in \Lambda$ and $y \notin \tilde{\alpha}$, $\alpha y \in \Lambda$ if and only if $(\rho/\tilde{\alpha})(y) = 1$. But more importantly, the representation defines a global submodular structure over the greedoid, which applies to both those sets corresponding to and those which do not correspond to supports of feasible words. This submodular structure is a crucial property given by matroid rank functions which is lost when generalizing from matroids to greedoids. The concept of representation is thus notable because it restores this feature. In [44], the authors define a subclass called *strong polymatroid greedoids* by those guaranteeing the function $\check{\rho}$, where $\check{\rho}(X)$ is the height of a minimal flat support containing X in the lattice $(\kappa(\Lambda/\sim), \sqsubseteq)$, i.e. $\check{\rho}(X) \triangleq \min\{|\alpha| \mid X \subseteq \kappa[\alpha]\}$ is a polymatroid rank function representing the greedoid. When true, $\check{\rho}$ is called the *greatest representation* because it can be shown to be coordinatewise larger than all other representations. The authors give a combinatorial description of such greedoids: a greedoid is *optimistic* whenever for every basic word $y_1 \dots y_R$ and nonloop $z \in \Sigma$ there exists an index i making $y_1 \dots y_{i-1}z \in \Lambda$. We remark that all polymatroid greedoids are optimistic [44, 24]. Then, Λ is a strong polymatroid greedoid if and only if it is an optimistic interval greedoid whose flat supports are closed under intersection [44], i.e. $\kappa(F) \cap \kappa(F') = \kappa(F \sqcap F')$ for all $F, F' \in \Lambda/\sim$. This shows the existence of a closure operator $\text{cl}(X) \triangleq \bigcap \{\kappa[\alpha] \mid X \subseteq \kappa[\alpha]\}$ which preserves the greedoid structure. Notice, cl also simplifies the definition of $\check{\rho}$ to make $\check{\rho}(X)$ the height of $\text{cl}(X)$ in $(\kappa(\Lambda/\sim), \sqsubseteq)$. By inspection then $\sigma_{\check{\rho}} = \text{cl}$, and so the greatest representation gives us functional access to this closure system. For this reason, $\check{\rho}$ and cl are meaningful tools for algorithm design.

4 Matroidal Prerequisite Systems

Now we define a matroidal prerequisite system. This is an interaction between a poset and a matroid, with the poset encoding prerequisites and the matroid encoding a substitutability structure, combined with a compatibility condition (we use this naming to informally refer to Eq. 3) ensuring this interaction is well-behaved.

► **Definition 3** (Matroidal Prerequisite System (MPS)). *Fix a finite alphabet Σ , and let $(\Sigma, \mathcal{M})$ and $(\Sigma, \rightsquigarrow)$ be a (loopless) matroid and poset respectively. Define $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ as the set of simple words $x_1 \dots x_\ell \in \Sigma^*$ with $\{x_1, \dots, x_\ell\} \in \mathcal{M}$ and,*

$$y \rightsquigarrow x_i \implies y \in \sigma_{\mathcal{M}}(\{x_1, \dots, x_{i-1}\}) \text{ or } \sigma_{\mathcal{M}}(\{x_1, \dots, x_{i-1}, y\}) = \sigma_{\mathcal{M}}(\{x_1, \dots, x_{i-1}, x_i\}),$$

for all $i \in \{1, \dots, \ell\}$. Then, $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ is a matroidal prerequisite system if and only if,

$$\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow) \text{ and } z \in \sigma_{\mathcal{M}}(\tilde{\alpha}) \implies (\forall y \rightsquigarrow z) y \in \sigma_{\mathcal{M}}(\tilde{\alpha}). \quad (3)$$

This is, $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ is words with independent supports such that every prerequisite of a letter y in a feasible word is either spanned by prefix ending at y (exclusive) in the matroid or is substitutable with y against that prefix. Call the words of $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ *feasible*. Then, the compatibility condition in Eq. 3 states that whenever a letter is covered by the matroid span

of a feasible word, so must all of its prerequisites. This has good semantic meaning with the problems we are interested in modeling (i.e. a decision which has been made redundant is such that all of its prerequisite decisions are redundant as well), and is required to verify $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ is an optimistic greedoid (which we need to use $\check{\rho}$, cl , and other concepts inducing correct behavior in our algorithms). We also implicitly assume the matroid is loopless so as to prevent complications or cumbersome notation caused by loops. This can be satisfied by simply removing letters with matroid rank zero from the groundset. Finally, as discussed in Section 1, this abstracts matroids and poset antimatroids by considering the MPS defined over an empty order and free matroid, respectively.

Before continuing, we list a few properties. Firstly, every basic word of $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ is always supported on a basic set (largest independent set) in $\mathcal{M}$. This is because, for all $\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$, any minimal (w.r.t. “ $\rightsquigarrow$ ”) letter $y \in \Sigma \setminus \sigma_{\mathcal{M}}(\tilde{\alpha})$ is such that minimality guarantees all its prerequisites are covered by $\sigma_{\mathcal{M}}(\tilde{\alpha})$. Moreover, $y \notin \sigma_{\mathcal{M}}(\tilde{\alpha})$ also implies $\tilde{\alpha} + y$ is independent, and so $\alpha y \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$. Such a y always exists whenever $\sigma_{\mathcal{M}}(\tilde{\alpha}) \neq \Sigma$, which occurs exactly when $\tilde{\alpha}$ is not a basic set of the matroid. Secondly, this implies every MPS defined by a loopless matroid is also loopless. Specifically, since $\sigma_{\mathcal{M}}(\tilde{\alpha}) = \Sigma$ for every basic word α , for all $y \in \Sigma$ there exists an α -prefix $\alpha'z$ such that $y \in \sigma_{\mathcal{M}}(\tilde{\alpha}' + z) \setminus \sigma_{\mathcal{M}}(\tilde{\alpha}')$. Then, since y is $\tilde{\alpha}'$ -substitutable with all its prerequisites by the compatibility condition (Eq. 3), it follows that $\alpha'y \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$. In what comes, we show an MPS is optimistic, and so this looplessness will mean that given any basic word and letter $y \in \Sigma$ we can assert the existence of a prefix possessing y as a continuation.

As a technical example, let $\mathcal{U}_n^R$ be a *uniform matroid* of rank R . That is, every set of cardinality at most R is independent. $\mathcal{L}(\mathcal{U}_n^R, \rightsquigarrow)$ always satisfies the compatibility condition, regardless of the prerequisite order $(\Sigma, \rightsquigarrow)$. This is because the only independent sets with span not equal to themselves are basic. So, basic words in $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ are the only feasible words whose corresponding matroid span contains any letters not from their support. Therefore, for all $\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$ which are not basic we have $z \in \sigma_{\mathcal{M}}(\tilde{\alpha})$ implies $z \in \tilde{\alpha}$, and so all prerequisites $y \rightsquigarrow z$ are contained in $\sigma_{\mathcal{M}}(\tilde{\alpha})$ by MPS feasibility, while if α is basic then it is supported on a basic set of $\mathcal{U}_n^R$ and $\sigma_{\mathcal{M}}(\tilde{\alpha}) = \Sigma$ trivially makes the conclusion of Eq. 3 true. This makes $\mathcal{L}(\mathcal{U}_n^R, \rightsquigarrow)$ a matroidal prerequisite system. We can use this to model cardinality constrained optimization over a poset antimatroid. This is equivalent to adding a cardinality constraint on the previously studied maximum closure over a directed acyclic graph [38], and restricting to unit-weights for previously studied poset constrained knapsacks [19, 41, 20].

► **Example 4** (Cardinality Constrained Optimization on a Poset Antimatroid). Let $(\Sigma, \rightsquigarrow)$ be a poset with $|\Sigma| = n$, $w : \Sigma \rightarrow \mathbb{R}$ a nonnegative weight function, and $k \leq n$ an integer. Our task is to find an ideal of size k of maximum cumulative weight, or equivalently finding a k -length prefix of a linear extension with maximum cumulative weight. Add an additional “sentinel letter” t by letting $\Sigma' = \Sigma + t$, and consider the matroid $(\Sigma', \mathcal{U}_{n+1}^{k+1})$. Extend the order to $(\Sigma', \rightsquigarrow')$ by letting $y \rightsquigarrow z \implies y \rightsquigarrow' z$ and $y \rightsquigarrow' t$ for all $y \in \Sigma$. This is, the derived order inherits the relations given by “ $\rightsquigarrow$ ” while making every letter lesser than the sentinel. Finally, make the derived weight function $w' : \Sigma' \rightarrow \mathbb{R}$ by letting $w'(y) = w(y)$ for all $y \in \Sigma$, and $w'(t) = \max_{y \in \Sigma} w(y) + 1$. It can be seen that any word of maximum weight over the MPS $\mathcal{L}(\mathcal{U}_{n+1}^{k+1}, \rightsquigarrow')$ is a basic word ending in t . It follows that the first k letters of such a word will be an ideal of $(\Sigma, \rightsquigarrow)$ of size k maximizing $X \mapsto \sum_{y \in X} w(y)$.

As a less technical example, we discuss how our MPS can be used to enrich the constraint structure in sensor placement problems.

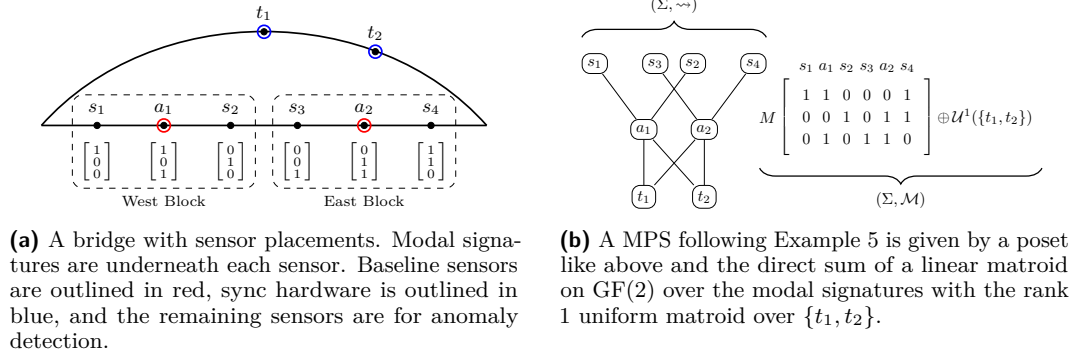


Figure 1 Following Example 5, modal signatures of the baseline sensor of each block must be spanned before an anomaly detection sensor is placed in that block. Furthermore, we must place sync hardware before any other sensor. This leads to the poset structure on the right. For budget reasons, we can only place one sync hardware. A linear matroid encodes the substitution structure of the signatures as well as a linear independence condition in the measurements, so the matroid is given by a direct sum like on the right. To save space, verifying the compatibility condition is deferred to the full version. Some examples of basic words are $t_1 a_1 s_1 s_2$, $t_1 a_2 s_4 s_1$, and $t_2 a_1 a_2 s_1$.

► **Example 5** (Sensor Networks for Anomaly Detection). Sensor placement to maximize mutual information [26] is a classic instance of monotone submodular maximization, with particularly natural matroid constraints encoding budget, network interference, etc. (see [39, 8] for example). One task is using sensors to detect anomalies; for example the presence of errant vibration frequencies indicates damage in a structure [1]. However, it is not possible to interpret anomalies without a frame of reference given by baseline measurements. This creates a prerequisite structure, where modes corresponding to a baseline must be spanned before anomaly detection can occur. Other operational prerequisites may coexist, like requiring synchronization hardware is placed before other hardware to ensure a correct boot procedure.

An instance of detecting anomalous frequencies on a bridge is shown in Figure 1. A modal signature is a binary vector encoding the capabilities of each sensor, and a linear matroid in combination with a poset encodes the requirement that baseline sensor modal signatures in each block must be spanned before anomaly detecting sensors in those blocks can be placed, along with the requirement that clock synchronization hardware is placed before any sensor. An interesting feasible word is $t_1 a_2 s_4 s_1$. Even though the baseline sensor a_1 was never placed in the East block, we still place s_1 after $t_1 a_2 s_4$ since the modal signature of a_1 is in the span of $\{a_2, s_4\}$. This is because the functionality of a_1 is redundant in the presence of a_2 and s_4 .

In Section 4.1, we examine the greedoid theory of an MPS. In particular, we show every MPS can be described as a strong polymatroid greedoid and vice versa. The utility of this is best seen in Section 4.3, where we use the greatest representation and closure operator of strong polymatroid greedoids to give procedures for “filling-in” and “projecting” partial optimization solutions on to the basic words of an MPS. This is Lemma 11, which will be the foundation of our algorithms in Sections 5 and 6. Finally, since we use the greatest representation in our algorithms, we verify that it can be computed efficiently when given a matroid rank oracle in Section 4.2. This is because, even though our algorithms are described in terms of strong polymatroid greedoid concepts, our previous examples show that the MPS is a widely applicable and interpretable constraint model. And so, while strong polymatroid greedoids give the mechanisms for our algorithms, an MPS grounds our motivations for designing them in the first place. Therefore, we must verify that given an MPS it is possible to efficiently compute strong polymatroid greedoid concepts under reasonable assumptions.

4.1 Greedoid Theoretic Aspects

We now develop the greedoid theory of an MPS. In particular, we will show that a greedoid is a strong polymatroid greedoid if and only if there exists a matroid and poset giving its feasible words as an MPS. We first verify an MPS is an optimistic interval greedoid.

► **Proposition 6.** *Let $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ be a matroidal prerequisite system. Then, $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ is an optimistic interval greedoid.*

Proof. From the definition $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ is nonempty and hereditary. Let $\alpha, \beta \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$ with $|\beta| > |\alpha|$. Select the first z_1 appearing in $\tilde{\beta}$ with $z_1 \notin \sigma_{\mathcal{M}}(\tilde{\alpha})$. We show αz_1 is feasible. Well, $z_1 \notin \sigma_{\mathcal{M}}(\tilde{\alpha})$ implies $\tilde{\alpha} + z_1$ is independent. And, if $\beta' z_1$ is the β -prefix ending at z_1 , we see $\tilde{\beta}' \subseteq \sigma_{\mathcal{M}}(\tilde{\alpha})$ by construction, thus $\sigma_{\mathcal{M}}(\tilde{\beta}') \subseteq \sigma_{\mathcal{M}}(\tilde{\alpha})$. So, for every $y \rightsquigarrow z_1$ such that $y \notin \sigma_{\mathcal{M}}(\tilde{\alpha})$, it follows that $y \notin \sigma_{\mathcal{M}}(\tilde{\beta}')$. Since $\beta' \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$, as we have already shown the hereditary property, it must be that y and z_1 are β' -substitutable. Hence,

$$\sigma_{\mathcal{M}}(\tilde{\alpha} + y) = \sigma_{\mathcal{M}}(\tilde{\alpha} \cup \sigma_{\mathcal{M}}(\tilde{\beta}' + y)) = \sigma_{\mathcal{M}}(\tilde{\alpha} \cup \sigma_{\mathcal{M}}(\tilde{\beta}' + z_1)) = \sigma_{\mathcal{M}}(\tilde{\alpha} + z_1),$$

by $\sigma_{\mathcal{M}}(\tilde{\beta}') \subseteq \sigma_{\mathcal{M}}(\tilde{\alpha})$, and so y and z_1 are $\tilde{\alpha}$ -substitutable. Thus $\alpha z_1 \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$. If $|\beta| - |\alpha| > 1$, one can repeat this argument on the first letter z_2 in β with $z_2 \notin \sigma_{\mathcal{M}}(\tilde{\alpha} z_1)$ to show $\alpha z_1 z_2 \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$ as well. Repeating in this way $|\beta| - |\alpha| - 2$ more times shows the existence of the $k = |\beta| - |\alpha|$ length β -subword $z_1 \dots z_k$ such that $\alpha z_1 \dots z_k \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$. And so, $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ is an interval greedoid.

Now, let $x_1 \dots x_R \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$ be basic. Select $y \in \Sigma$ and let i be the smallest index with $y \in \sigma_{\mathcal{M}}(\{x_1, \dots, x_i\})$. By Steinitz-Maclane exchange property, x_i and y are $\{x_1, \dots, x_{i-1}\}$ -substitutes. Well, for every $z \rightsquigarrow y$ with $z \notin \sigma_{\mathcal{M}}(\{x_1, \dots, x_{i-1}\})$, the compatibility condition and the observation that x_i and y are $\{x_1, \dots, x_{i-1}\}$ -substitutes combine for,

$$y \in \sigma_{\mathcal{M}}(\{x_1, \dots, x_{i-1}, x_i\}) \implies z \in \sigma_{\mathcal{M}}(\{x_1, \dots, x_{i-1}, y\}).$$

So, $x_1 \dots x_{i-1} y \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$, and optimism follows. ◀

Now, we make observations about the interaction between the flat supports, matroid, and prerequisite order. Specifically, the next lemma first verifies flat supports are always $r_{\mathcal{M}}$ -closed and ideals of $(\Sigma, \rightsquigarrow)$. This then shows the flat supports are closed under intersection.

► **Lemma 7.** *Let $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ be a matroidal prerequisite system. The following are all true:*

1. *For all $\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$, $\sigma_{\mathcal{M}}(\tilde{\alpha}) = \kappa[\alpha]$,*
2. *Every flat support is an ideal in the prerequisite order $(\Sigma, \rightsquigarrow)$,*
3. *The flat supports are closed under intersection.*

To prove the first item, we need the following statement which follows from the semimodularity of the flats of interval greedoids. See [2] for proof.

► **Lemma 8 ([2]).** *Let Λ be an interval greedoid, and select nonempty words $\alpha x \in \Lambda$ and $\alpha y \in \Lambda$. If $\alpha x \not\rightsquigarrow \alpha y$, then $\alpha xy \in \Lambda$ and $\alpha yx \in \Lambda$.*

Proof of Lemma 7. For what follows, recall Proposition 6 shows $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ is an optimistic interval greedoid. Let $\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$ and select $y \in \sigma_{\mathcal{M}}(\tilde{\alpha})$. Since supports of feasible words are independent, there exists no word in the contraction $\mathcal{L}(\mathcal{M}, \rightsquigarrow)/\alpha$ containing y in its support. So by optimism (since every feasible word can be extended into a basic word), there exists an α -prefix $\alpha' z$ with $y \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)/\alpha'$. Let $\alpha' z$ be a longest such prefix, so that $\alpha' y, \alpha' z \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$ while neither $\alpha' yz$ or $\alpha' zy$ are feasible. By contrapositive of Lemma 8, it

follows that $\alpha'y \sim \alpha'z$. Because our construction makes $[\alpha'z] \subseteq [\alpha]$, it follows that $[\alpha'y] \subseteq [\alpha]$. So, there exists $\beta \in \Sigma^*$ making $\alpha'y\beta \in [\alpha]$, thus $y \in \kappa[\alpha]$. This shows $\sigma_{\mathcal{M}}(\tilde{\alpha}) \subseteq \kappa[\alpha]$. Now, select $z' \notin \sigma_{\mathcal{M}}(\tilde{\alpha})$. Then, there exists $\beta \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)/\alpha$ with $z' \in \tilde{\beta}$: There must exist a minimal $y' \in \downarrow z' \setminus \sigma_{\mathcal{M}}(\tilde{\alpha})$ w.r.t. “ $\rightsquigarrow$ ”, and this minimal y' must be a continuation of α since its prerequisites are covered by $\sigma_{\mathcal{M}}(\tilde{\alpha})$. Therefore, if $\alpha z' \notin \mathcal{L}(\mathcal{M}, \rightsquigarrow)$, we can append y' onto α . If we continue this process, eventually z' will be substitutable with every prerequisite that hasn't been covered, at which point z' will be a continuation. This makes $z' \notin \kappa[\alpha]$. So, as we've shown $\sigma_{\mathcal{M}}[\alpha] \subseteq \kappa[\alpha]$ and $\Sigma \setminus \sigma_{\mathcal{M}}(\tilde{\alpha}) \subseteq \Sigma \setminus \kappa[\alpha]$, it follows that $\sigma_{\mathcal{M}}(\tilde{\alpha}) = \kappa[\alpha]$. This gives Lemma 7.1. Finally, we note that our compatibility condition in Eq. 3 makes $\sigma_{\mathcal{M}}(\tilde{\alpha})$ equal to an ideal in $(\Sigma, \rightsquigarrow)$. Since we've shown $\kappa[\alpha] = \sigma_{\mathcal{M}}(\tilde{\alpha})$, this gives Lemma 7.2.

This description of the flat supports is convenient for showing they are closed under intersection. In particular, since ideals of a poset are always closed under intersection, it follows that $\kappa(F) \cap \kappa(F')$ is an ideal on $(\Sigma, \rightsquigarrow)$ for all $F, F' \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)/\sim$. It is also true that $\kappa(F \sqcap F')$ and $\kappa(F) \cap \kappa(F')$ are closed in $\mathcal{M}$. Now, observe that the inclusion $\kappa(F \sqcap F') \subseteq \kappa(F) \cap \kappa(F')$ is always true by $(\Lambda/\sim, \sqsubseteq) \cong (\kappa(\Lambda/\sim), \subseteq)$, so for 2. to hold we need only verify the reverse inclusion. By way of contradiction, suppose there exists $z \in (\kappa(F) \cap \kappa(F')) \setminus \kappa(F \sqcap F')$. Without loss of generality, let z be a minimal letter in this difference w.r.t. “ $\rightsquigarrow$ ”. Because $\kappa(F \sqcap F')$ is an ideal, minimality implies $y \in \kappa(F \sqcap F')$ for all $y \rightsquigarrow z$. Therefore, for all $\mu \in F \sqcap F'$, every prerequisite of z is covered by the span $\sigma_{\mathcal{M}}(\tilde{\mu})$ since $\sigma_{\mathcal{M}}(\tilde{\mu}) = \kappa(F \sqcap F')$ by 1. This makes z a continuation of words in $F \sqcap F'$, hence $F \sqcap F' \prec [\mu z]$. But, $z \in \kappa(F) \cap \kappa(F')$ implies $[\mu z]$ is lesser than both F and F' . One (of many) ways to see why is because if we select a word $\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)/\mu$ making $\mu\alpha \in F$, then by exchange there is an α -subword α' of length $|\alpha| - 1$ such that $\mu z \alpha'$ is feasible. This construction makes both $\tilde{\mu z \alpha'} \subseteq \kappa(F)$, and the flats F and $[\mu z \alpha']$ the same height in $(\Lambda/\sim, \sqsubseteq)$. So, by $(\Lambda/\sim, \sqsubseteq) \cong (\kappa(\Lambda/\sim), \subseteq)$, this makes $\mu z \alpha' \in F$. A symmetric argument applies to show $[\mu z] \subseteq F'$ as well. Therefore $[\mu z] \subseteq F \sqcap F'$. Yet, this contradicts $F \sqcap F' \prec [\mu z]$, so $\kappa(F) \cap \kappa(F') = \kappa(F \sqcap F')$. So we've shown Lemma 7.3. ◀

Therefore, by Lemma 7.3 and Proposition 6, every MPS is a strong polymatroid greedoid by the description from [44]. For logical completeness, we give a construction encoding every strong polymatroid greedoid as an MPS. Because the forthcoming results only use the direction already proven, we defer this to the full version to satisfy space constraints.

► **Theorem 9.** *Let Λ be a greedoid on the alphabet Σ . There exists a matroid $(\Sigma, \mathcal{M})$ and compatible poset $(\Sigma, \rightsquigarrow)$ such that Λ equals the words of the matroidal prerequisite system $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ if and only if Λ is a strong polymatroid greedoid.*

4.2 Computing the Greatest Representation

Our algorithms use the greatest representation $\check{\rho}$. To ensure this is well-founded, we show $\check{\rho}$ can be evaluated via a single rank query on the underlying matroid. Specifically, the greatest representation of an MPS $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ is the composition of the matroid rank and ideal operator on $(\Sigma, \rightsquigarrow)$. Note that ideals can be computed efficiently in $\text{poly}(|\Sigma|)$ -time by repeated ordered comparisons.

► **Proposition 10.** *Let $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ be an MPS. Then, $\check{\rho} = r_{\mathcal{M}} \circ \downarrow$.*

Proof. Fix $X \subseteq \Sigma$. We verify $\check{\rho}(X) = r_{\mathcal{M}}(\downarrow X)$. Perform the following procedure: Begin with a word $\alpha = \epsilon$. Then, in each step, select a smallest $z \in \downarrow X \setminus \tilde{\alpha}$ satisfying $r_{\mathcal{M}}(\tilde{\alpha} + z) > r_{\mathcal{M}}(\tilde{\alpha})$. Terminate once no such z exists. See that $\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$ because, fixing z to a selection made in any round, minimality makes every $y \rightsquigarrow z$ spanned by the round's value of α . Moreover,

it is the case that $(r_{\mathcal{M}}/\tilde{\alpha})(\downarrow X) = 0$ upon termination. So, with α being the value at the end of the final round, we have $\downarrow X \subseteq \sigma_{\mathcal{M}}(\tilde{\alpha})$. Yet combining this with $\tilde{\alpha} \subseteq \downarrow X$ makes $r_{\mathcal{M}}(\tilde{\alpha}) \leq r_{\mathcal{M}}(\downarrow X) \leq r_{\mathcal{M}}(\tilde{\alpha})$, so $r_{\mathcal{M}}(\downarrow X) = |\alpha|$ since $\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$ implies $\tilde{\alpha} \in \mathcal{M}$. Now, take note that $\downarrow X \subseteq \sigma_{\mathcal{M}}(\tilde{\alpha})$ also implies that no $\beta \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)/\alpha$ satisfies $\tilde{\beta} \cap \downarrow X \neq \emptyset$, as the supports of feasible words must be independent. This makes $(\check{\rho}/\tilde{\alpha})(y) \neq 1$ for all $y \in \downarrow X$ in particular. By optimism (see Proposition 6) it follows that there exists an α -prefix $\alpha^{(y)}z$ for each $y \in \downarrow X$ such that $\alpha^{(y)}y$ is feasible. So, $(\check{\rho}/\tilde{\alpha}^{(y)})(y) = 1$. But by the law of diminishing returns and $(\check{\rho}/\tilde{\alpha})(y) \neq 1$ it follows that $1 = (\check{\rho}/\tilde{\alpha}^{(y)})(y) > (\check{\rho}/\tilde{\alpha})(y)$ for all $y \in \downarrow X$. And so, as $\check{\rho}$ is integral, we have $(\check{\rho}/\tilde{\alpha})(y) = 0$ for all $y \in \downarrow X$. This makes $\downarrow X \subseteq \text{cl}(\tilde{\alpha})$. Therefore $\check{\rho}(\tilde{\alpha}) \leq \check{\rho}(\downarrow X) \leq \check{\rho}(\tilde{\alpha})$, and so $\check{\rho}(\downarrow X) = \check{\rho}(\tilde{\alpha})$. But, because flat supports are ideals by Lemma 7.2, we see $\downarrow X \subseteq \text{cl}(X)$, and so $\check{\rho}(X) \leq \check{\rho}(\downarrow X) \leq \check{\rho}(X)$ by $\text{cl} = \sigma_{\check{\rho}}$. Therefore, $\check{\rho}(X) = \check{\rho}(\tilde{\alpha})$, and since $\check{\rho}$ is a representation we have $\check{\rho}(\tilde{\alpha}) = |\alpha|$. Hence we've found $r_{\mathcal{M}}(\downarrow X) = |\alpha|$ and $\check{\rho}(X) = |\alpha|$. ◀

4.3 Projection and Fill-In

In Sections 5 we present algorithms based on projecting a relaxed solution back onto the words of an MPS, while in Section 6 our algorithms build words through repeated insertions eventually converging to a feasible word. Both ideas require a description of what sets of letters, which might not support a feasible word, can be “filled” into a feasible word. In the following, we show that words $x_1 \dots x_\ell$ such that $(\check{\rho}/\{x_1, \dots, x_{i-1}\})(x_i) > 0$ for all $i \in \{1, \dots, \ell\}$ satisfy this. We note this is a special property of strong polymatroid greedoids.

► **Lemma 11.** *Suppose Λ is a strong polymatroid greedoid. Then, for all simple $x_1 \dots x_k \in \Sigma^*$, if $(\check{\rho}/\{x_1, \dots, x_{i-1}\})(x_i) > 0$ for all $i \in \{1, \dots, k\}$, there exists a set of letters $\{y_{i,1}, \dots, y_{i,\ell_i}\}$ with $\ell_i = (\check{\rho}/\{x_1, \dots, x_{i-1}\})(x_i) - 1$ inducing the feasible word,*

$$y_{1,1} \dots y_{1,\ell_1} x_1 y_{2,1} \dots y_{2,\ell_2} x_2 \dots y_{k,1} \dots y_{k,\ell_k} x_k \in \Lambda.$$

Proof. By $(\check{\rho}/\{x_1, \dots, x_{i-1}\})(x_i) > 0$ for all $i \in \{1, \dots, k\}$, the existence of the strictly ascending chain $\emptyset \subsetneq \text{cl}(\{x_1\}) \subsetneq \text{cl}(\{x_1, x_2\}) \subsetneq \dots \subsetneq \text{cl}(\{x_1, \dots, x_k\})$ of flat supports in $(\kappa(\Lambda/\sim), \subseteq)$ follows. Then, select letters $\{y_{i,1}, \dots, y_{i,\ell_i}\}$ for each x_i inducing a covering chain beginning at $\text{cl}(\{x_1, \dots, x_{i-1}\})$ and ending at $\text{cl}(\{x_1, \dots, x_i\})$ like,

$$\begin{aligned} \text{cl}(\{x_1, \dots, x_{i-1}\}) &\prec \text{cl}(\{x_1, \dots, x_{i-1}, y_{i,1}\}) \prec \dots \prec \text{cl}(\{x_1, \dots, x_{i-1}, y_{i,1}, \dots, y_{i,\ell_i}\}), \\ &\prec \text{cl}(\{x_1, \dots, x_{i-1}, y_{i,1}, \dots, y_{i,\ell_i}, x_i\}) = \text{cl}(\{x_1, \dots, x_i\}). \end{aligned} \quad (4)$$

This is always possible because cl is a closure operator. In particular, since $\sigma_{\check{\rho}} = \text{cl}$, having already selected $y_{i,1}, \dots, y_{i,j-1}$, one takes an arbitrary $y_{i,j}$ satisfying both of,

$$\begin{cases} (\check{\rho}/\{x_1, \dots, x_{i-1}, y_{i,1}, \dots, y_{i,j-1}\})(y_{i,j}) = 1, \\ \text{cl}(\{x_1, \dots, x_{i-1}, y_{i,1}, \dots, y_{i,j-1}\} + y_{i,j}) \subseteq \text{cl}(\{x_1, \dots, x_i\}), \end{cases} \quad (5)$$

whenever $(\check{\rho}/\{x_1, \dots, x_{i-1}, y_{i,1}, \dots, y_{i,j-1}\})(x_i) \neq 1$, and terminates otherwise. A letter $y_{i,j}$ satisfying the conditions in Eq. 5 must exist when $(\check{\rho}/\{x_1, \dots, x_{i-1}, y_{i,1}, \dots, y_{i,j-1}\})(x_i) \neq 1$, as there is a flat support below $\text{cl}(\{x_1, \dots, x_i\})$ covering $\text{cl}(\{x_1, \dots, x_{i-1}, y_{i,1}, \dots, y_{i,j-1}\})$ in $(\kappa(\Lambda/\sim), \subseteq)$, and any letter in the difference of this cover and $\text{cl}(\{x_1, \dots, x_{i-1}, y_{i,1}, \dots, y_{i,j-1}\})$ will satisfy the conditions in Eq. 5. Now, recall that $\check{\rho}(X)$ is defined as the height of the least flat support containing X (recall flat supports are closed under intersection, so a least such flat is unique). So, $\check{\rho}$ grades the cl -image. Then, the covering chain structure from Eq. 4 and $\sigma_{\check{\rho}} = \text{cl}$ makes,

$$(\check{\rho}/\{x_1, \dots, x_{i-1}, y_{i,1}, \dots, y_{i,j-1}\})(y_{i,j}) = 1, \quad \forall j \in \{1, \dots, \ell_i\}. \quad (6)$$

This means, for example, that $y_{1,1} \dots y_{1,\ell_1} x_1 \in \Lambda$ by the definition of representation in polymatroid greedoids. Then, since selecting like Eq. 4 makes $\text{cl}(\{y_{1,1}, \dots, y_{1,\ell_1}, x_1\}) = \text{cl}(\{x_1\})$, applying the same argument using Eq. 6 for $i = 2$ shows $y_{1,1} \dots y_{1,\ell_1} x_1 y_{2,1} \dots y_{2,\ell_2} x_2 \in \Lambda$. One repeats this procedure up to $i = k$ to get the result. $\blacktriangleleft$

5 Scaffolding Solutions via Polymatroid Relaxations

Observe, every polymatroid greedoid is such that the characteristic vectors of its basic words are base vectors of any of its polymatroid representations. To see why, note that Edmonds' algorithm [11] over the base polytope always returns an extreme point. Let $y_1 \dots y_R \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$ be any basic word of an MPS with greatest representation $\check{\rho}$. Then, a weight function $w : \Sigma \rightarrow \mathbb{R}$ such that $w(y_i) = R - (i - 1)$ for all $i \in \{1, \dots, R\}$ and $w(z) = 0$ for $z \notin \{y_1, \dots, y_R\}$ will make Edmonds' algorithm examine coordinates in the order dictated by the basic word. Because $\check{\rho}$ is a representation, we see that $(\check{\rho}/\{y_1, \dots, y_{i-1}\})(y_i) = 1$ for all i , and Lemma 11 certifies that this ends at y_R , as the existence of $z \in \Sigma$ with $(\check{\rho}/\{y_1, \dots, y_R\})(z) > 0$ would imply $y_1 \dots y_R$ is not basic. Therefore, Edmonds' algorithm will increase each coordinate from $\{y_1, \dots, y_R\}$ to exactly one and terminate on the characteristic vector of the support of the basic word.

In what follows we will examine nonnegative additive maximization and monotone submodular maximization algorithms based on using the greatest representation as a relaxation. Specifically, Section 5.1 will first describe a method for additive maximization in Algorithm 1 which is based on computing a relaxed solution via Edmonds' algorithm and projecting onto a basic word using Lemma 11. In Section 5.2 we approach submodular maximization using the continuous greedy algorithm [5] to compute a set of weights forming an additive underestimation of the objective, which we give to our Algorithm 1 to compute a solution on the MPS. The resulting guarantees will vary in largest matroid rank of a principal ideal Δ .

5.1 Additive Maximization

Let $w : \Sigma \rightarrow \mathbb{R}_+$ be a nonnegative weight function. Our task is to find a feasible word α of a MPS $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ maximizing $\alpha \mapsto \sum_{y \in \tilde{\alpha}} w(y)$. Let $v^* \in \mathbb{R}^\Sigma$ be base vector of $\check{\rho}$ computed by Edmonds' algorithm, and sort the nonzero coordinates $\{x_1, x_2, \dots\} = \text{supp}(v^*)$ so that $w(x_1) \geq w(x_2) \geq \dots$. This is, we sort the support of v^* in order of the coordinate increases made by Edmonds' algorithm. Since $x_i \in \text{supp}(v^*)$, it follows that $(\check{\rho}/\{x_1, \dots, x_{i-1}\})(x_i) > 0$ for all $i \in \{1, \dots, |\text{supp}(v^*)|\}$. This satisfies the premise of Lemma 11, and so perform a fill-in procedure like in the proof of Lemma 11 to obtain a feasible word containing $\text{supp}(v^*)$.

This idea is shown in Algorithm 1. We first compute a ‘‘scaffold’’ v^* for the solution using Edmonds' greedy algorithm on $\check{\rho}$ in line 1, whose support we then sort in line 2. Then, we use the same arbitrary fill-in scheme presented in the proof of Lemma 11, in particular Eq. 5, to obtain a feasible word α such that $\text{supp}(v^*) \subseteq \tilde{\alpha}$ in lines 4 through 8. Note that the while loop cannot terminate with $(\check{\rho}/\tilde{\alpha})(x_i) = 0$: if a selected fill-in letter y with $(\check{\rho}/\tilde{\alpha})(y) = 1$ made $x_i \in \text{cl}(\tilde{\alpha} + y)$, then we would have $(\check{\rho}/\tilde{\alpha})(x_i) \leq 1$ which contradicts the loop condition.

Recall, in Section 1 we defined Δ as the maximum matroid rank of a principal ideal of $(\Sigma, \rightsquigarrow)$. By the law of diminishing returns, it follows that each coordinate of v^* is at most Δ . When comparing v^* and $\chi_{\tilde{\alpha}}$, it follows that any entry in v^* is at most a Δ -factor smaller in $\chi_{\tilde{\alpha}}$. This idea shows Algorithm 1 computes a Δ -approximation.

Algorithm 1 Scaffolded Greedy with Arbitrary Fill-In.

```

1  $v^* \leftarrow$  solution from Edmonds' algorithm [11] on  $\check{\rho}$  with weights  $w$ 
2 select  $x_1 \dots x_k \in \text{supp}(v^*)!$  with  $w(x_1) \geq w(x_2) \geq \dots \geq w(x_k)$ 
3  $\alpha \leftarrow \epsilon$ 
4 for  $i \in \{1, \dots, k\}$  do
5   while  $(\check{\rho}/\tilde{\alpha})(x_i) > 1$  do
6     select  $y \in \Sigma$  with  $(\check{\rho}/\tilde{\alpha})(y) = 1$  and  $\tilde{\alpha} + y \subseteq \text{cl}(\{x_1, \dots, x_i\})$ 
7      $\alpha \leftarrow \alpha y$ 
8    $\alpha \leftarrow \alpha x_i$ 
9 return  $\alpha$ 

```

► **Theorem 12.** Let $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ be a matroidal prerequisite system, $w : \Sigma \rightarrow \mathbb{R}_+$ a nonnegative weight function, and $\text{OPT} = \max_{\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)} \sum_{y \in \tilde{\alpha}} w(y)$. Algorithm 1 outputs $\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$ with $\sum_{y \in \tilde{\alpha}} w(y) \geq \Delta^{-1} \cdot \text{OPT}$.

Proof. Firstly, one sees that the fill-in (lines 4-8) is done in the same way as the proof of Lemma 11, so it is clear from our discussion at the start of the section that the resulting α is feasible. Moreover, it is basic as $\check{\rho}(\text{supp}(v^*)) = \check{\rho}(\Sigma)$ and $\text{cl}(\tilde{\alpha}) = \text{cl}(\text{supp}(v^*))$ makes $\check{\rho}(\tilde{\alpha}) = \check{\rho}(\Sigma)$. For each $x_i \in \text{supp}(v^*)$, by definition of Edmonds' algorithm (see [11]) we have $v^*(x_i) = (\check{\rho}/\{x_1, \dots, x_{i-1}\})(x_i)$. Therefore, for any choice of $y \in \text{supp}(v^*)$, we have $v^*(y) \leq \check{\rho}(y)$ by the law of diminishing returns. Of course $\check{\rho}(y) \leq \Delta$, and so,

$$\chi_{\tilde{\alpha}}(y) \geq \frac{1}{\Delta} \cdot v^*(y), \quad \forall y \in \text{supp}(v^*). \quad (7)$$

Then, let α_{OPT} be an optimal word. Recall by our discussion at the beginning of Section 5 that $\chi_{\alpha_{\text{OPT}}}$ is an extreme point of the base polytope, and so as Edmonds' algorithm is optimal over this constraint set we see $\text{OPT} = w^\top \chi_{\alpha_{\text{OPT}}} \leq w^\top v^*$. Combining this with Eq. 7 shows the desired $w^\top \chi_{\tilde{\alpha}} \geq \Delta^{-1} \cdot w^\top v^* \geq \Delta^{-1} \cdot \text{OPT}$. ◀

5.2 Monotone Submodular Maximization

For what comes, we require the following: Every integral polymatroid can be associated with a matroid via the *natural matroid* construction [16]. Specifically, define the ground set as,

$$E_{\check{\rho}} \triangleq \bigcup_{y \in \Sigma} \{(y, 1), \dots, (y, \check{\rho}(y))\}.$$

This just introduces $\check{\rho}(y)$ copies of each letter $y \in \Sigma$. This is equivalently seen as a multiset, so let the multiplicity vector of $S \subseteq E_{\check{\rho}}$ be given by $\mu_S(y) \triangleq |S \cap (\{y\} \times \{1, \dots, \check{\rho}(y)\})|$. Then, a matroid $M(\check{\rho})$ is defined by letting S be independent if and only if $\mu_S^\top \chi_X \leq \check{\rho}(X)$ for all $X \subseteq \Sigma$. The resulting rank function is,

$$r_{M(\check{\rho})}(S) \triangleq \min_{X \subseteq \Sigma} \check{\rho}(X) + \sum_{y \in \Sigma \setminus X} \mu_S(y),$$

which we note can be evaluated efficiently by submodular minimization algorithms [28, 42, 18, 36] over $X \mapsto \check{\rho}(X) + \sum_{y \in \Sigma \setminus X} \mu_S(y)$. This is a standard construction; see [16, 3] or Ch. 44 of [43] for more discussion.

Algorithm 2 Scaffolding with Weights Computed from Continuous Greedy.

```

1  $S \leftarrow$  continuous greedy with rounding on  $T \mapsto \sum_{k=1}^{\Delta} f(T(k))$  over  $(E_{\check{\rho}}, M(\check{\rho}))$ 
2 forall  $k \in \{1, \dots, \Delta\}$  do
3   select any  $y_1 y_2 \dots \in S(k)!$ 
4   forall  $i \in \{1, \dots, |S(k)|\}$  do
5      $w^{(k)}(y_i) \leftarrow (f/\{y_1, \dots, y_{i-1}\})(y_i)$ 
6   forall  $z \notin S(k)$  do
7      $w^{(k)}(z) \leftarrow 0$ 
8  $\bar{w} \leftarrow \Delta^{-1} \cdot \sum_{k=1}^{\Delta} w^{(k)}$ 
9  $\alpha \leftarrow$  scaffolded greedy on  $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$  with weights  $\bar{w}$  // Using Algorithm 1
10 return  $\alpha$ 

```

Now, let $f : 2^{\Sigma} \rightarrow \mathbb{R}$ be a monotone submodular function. Our next algorithm is based off of solving a monotone submodular maximization, using a derived objective function $g : 2^{E_{\check{\rho}}} \rightarrow \mathbb{R}$, over the natural matroid using the continuous greedy algorithm [5]. The corresponding solution then defines an additive function underestimating the objective f , which we feed into Algorithm 1 to obtain a word from the MPS. To define g , for $S \subseteq E_{\check{\rho}}$ let its k^{th} slice $S(k)$ be $S(k) \triangleq \{y \in \Sigma \mid (y, k) \in S\}$. Notice, as Δ is the maximum rank of a principal ideal and $\check{\rho} = r_{\mathcal{M}} \circ \downarrow$, it follows there are at most Δ slices to evaluate for any $S \subseteq E_{\check{\rho}}$. Then, define $g : 2^{E_{\check{\rho}}} \rightarrow \mathbb{R} : S \mapsto \sum_{k=1}^{\Delta} f(S(k))$. Note g is monotone submodular since it is a sum of monotone submodular functions on disjoint restricted domains.

Algorithm 2 begins by obtaining an approximate solution S to maximizing g over the natural matroid $(E_{\check{\rho}}, M(\check{\rho}))$ in line 1. Then, in lines 3 through 7 we compute a set of weights for each slice of S . Specifically, an arbitrary permutation $y_1 y_2 \dots$ is chosen from $S(k)!$ to make $w^{(k)}(y_i) = (f/\{y_1, \dots, y_{i-1}\})(y_i)$ for all $i \in \{1, \dots, |S(k)|\}$ and $w^{(k)}(z) = 0$ for $z \notin S(k)$. These weight functions are averaged together in line 8 to obtain the weight function $\bar{w}$. This is, in some sense, an additive estimate of f informed by a solution obtained from a proxy function. We then obtain a word of the MPS in line 9 by running Algorithm 1 with the weights defined by $\bar{w}$, and output that word. The resulting guarantee is that of continuous greedy with pipage rounding [5] degraded by a Δ^2 -factor due to the maneuvers involved in translating its solution to a feasible word.

► **Theorem 13.** *Let $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ be a matroidal prerequisite system, $f : 2^{\Sigma} \rightarrow \mathbb{R}$ a (normalized) monotone submodular function, and $\text{OPT} = \max_{\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)} f(\tilde{\alpha})$. Algorithm 2 outputs $\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$ with $\mathbb{E}[f(\tilde{\alpha})] \geq \Delta^{-2}(1 - 1/e - \delta) \cdot \text{OPT}$ in randomized $\text{poly}(|\Sigma|, 1/\delta)$ -time for $\delta > 0$.*

Proof. Let S be the independent set of $(E_{\check{\rho}}, M(\check{\rho}))$ approximately maximizing g , as computed in line 1. Then, $\bar{w}^\top \mu_S = \Delta^{-1} \cdot \sum_{k=1}^{\Delta} \sum_{y \in \Sigma} \mu_S(y) w^{(k)}(y)$. But, because lines 6-9 make $\sum_{y \in S(k)} w^{(k)}(y) = f(S(k))$, we find,

$$\frac{1}{\Delta} \cdot \sum_{k=1}^{\Delta} \sum_{y \in \Sigma} \mu_S(y) w^{(k)}(y) \geq \frac{1}{\Delta} \cdot \sum_{k=1}^{\Delta} \sum_{y \in S(k)} w^{(k)}(y) \geq \frac{1}{\Delta} \cdot \sum_{k=1}^{\Delta} f(S(k)) = \frac{g(S)}{\Delta},$$

hence $\bar{w}^\top \mu_S \geq \Delta^{-1} \cdot g(S)$.

Now, let v^* be the solution computed by Edmonds' algorithm over the base polytope of $\check{\rho}$ when given weights $\bar{w}$, and examine the final word α computed in line 9. Because no coordinate in v^* can exceed value Δ , $\sum_{y \in \tilde{\alpha}} \bar{w}(y) \geq \Delta^{-1} \bar{w}^\top v^*$ follows by the law of

■ **Algorithm 3** Marginal Adjusted Greedy for Additive Maximization.

```

1  $\alpha \leftarrow \epsilon$ 
2 while  $|\alpha| < r_{\mathcal{M}}(\Sigma)$  do
3    $\ell \leftarrow |\alpha|$ 
4    $x_1 \dots x_\ell \leftarrow \alpha$ 
5    $K_0 \leftarrow \emptyset$ 
6   for  $i \in \{1, \dots, \ell\}$  do  $K_i \leftarrow \text{cl}(\{x_1, \dots, x_i\})$ 
7   for  $i \in \{0, \dots, \ell\}$  do
8     if  $i < \ell$  then  $\Psi_i \leftarrow \{y \in \Sigma \mid K_i \subsetneq \text{cl}(K_i + y) \subsetneq K_{i+1}\}$ 
9     else  $\Psi_i \leftarrow \{y \in \Sigma \mid K_i \subsetneq \text{cl}(K_i + y) \subseteq \Sigma\}$ 
10     $(y^*, i^*) \leftarrow \arg \max_{y \in \Psi_i: 0 \leq i \leq \ell} \frac{w(y)}{(\bar{\rho}/K_i)(y)}$ 
11     $\alpha \leftarrow x_1 \dots x_{i^*} y^* x_{i^*+1} \dots x_\ell$ 
12 return  $\alpha$ 

```

diminishing returns. But, the natural matroid construction shows μ_S is in the base polytope, and so by optimality of v^* we see $\bar{w}^\top \mu_S \leq \bar{w}^\top v^*$. So, combining this with the insight of the last paragraph $\bar{w}^\top v^* / \Delta \geq \bar{w}^\top \mu_S / \Delta \geq g(S) / \Delta^2$. Then, noting that $\sum_{y \in X} w^{(k)}(y) \leq f(X)$ for all $k \in \{1, \dots, \Delta\}$ and $X \subseteq \Sigma$ by the law of diminishing returns, combining the last equation with $\sum_{y \in \tilde{\alpha}} \bar{w}(y) \geq \Delta^{-1} \bar{w}^\top v^*$ shows $f(\tilde{\alpha}) \geq \sum_{y \in \tilde{\alpha}} \bar{w}(y) \geq \Delta^{-2} g(S)$. To finish things up, let α_{OPT} be an optimal basic word, and $T_{\alpha_{\text{OPT}}} = \{(y, 1) \mid y \in \tilde{\alpha}_{\text{OPT}}\}$. See that $g(T_{\alpha_{\text{OPT}}}) = f(\tilde{\alpha}_{\text{OPT}}) + \sum_{k=2}^{\Delta} f(\emptyset) = f(\tilde{\alpha}_{\text{OPT}}) = \text{OPT}$. So, combining this with the guarantees of [5] and $f(\tilde{\alpha}) \geq \Delta^{-2} g(S)$ gives $\mathbb{E}[f(\tilde{\alpha})] \geq \Delta^{-2} \mathbb{E}[g(S)] \geq \Delta^{-2} (1 - 1/e - \delta) \cdot \text{OPT}$, which is the desired result. ◀

6 Marginal Adjusted Greedy Algorithms

The last algorithms were based upon “scaffolding” the solution with good letters from a relaxed solution, and using the fill-in procedure from Lemma 11 to project this solution to a characteristic vector corresponding to a basic word. However, the fill-in procedure was such that the letters were chosen only for feasibility, not with any heuristic guiding their value for a solution. In the next algorithms, we instead select letters using a greedy heuristic driven by the greatest representation. These algorithms will lead to approximation guarantees varying in the largest matroid connectivity $\lambda_{\max}$.

6.1 Additive Maximization

Algorithm 3 works by iteratively selecting letters to insert into a working solution (initially the empty word) in a way that both achieves good value and converges to a feasible word. In each round, the method first computes in line 6 the closed set K_i given by each prefix $x_1 \dots x_i$ of the current value of α , which we assume to be of length ℓ . We also make $K_0 = \emptyset$. Then, in line 8, for each $i \in \{0, 1, \dots, \ell\}$ we compute the set Ψ_i consisting of letters which strictly increase the closed set spanned by K_i to a closed set strictly less than K_{i+1} . The set Ψ_ℓ poses a boundary condition which is handled in line 9. We save such letters in the set Ψ_i because they can be inserted into α between x_i and x_{i+1} (or appended onto the beginning or end of α when i equals 0 or ℓ , respectively) while maintaining the invariant that the process will converge to a feasible word by way of Lemma 11. In particular, one can

view α as consisting of decisions we have already committed to (along with their relative ordering). Under this perspective, we use the closure structure and construction of the sets $\{\Psi_i\}_i$ to identify new decisions we can feasibly make while still preserving the structure of the decisions we have already committed to. This is shown in a technical way in Eq. 11 in the proof of Theorem 14. Finally, observe each letter is a member of at most one Ψ_i , because $y \in \Psi_i$ implies $y \in K_{i+1} \setminus K_i$ and the sequence $\{K_i\}_i$ forms a filtration.

Keeping this in mind, in line 10 we compute a marginal-adjusted weight for every letter $y \in \bigcup_{i=0}^{\ell} \Psi_i$. Specifically, when $y \in \Psi_i$, we divide $w(y)$ by the marginal return $(\check{\rho}/K_i)(y)$. This is because $(\check{\rho}/K_i)(y)$ is exactly the difference in grade between K_i and $\text{cl}(K_i + y)$, and so a large marginal return corresponds to a large jump in the closed set structure. If we commit to selecting y , then we commit to selecting a sequence of letters making a saturated chain between K_i and $\text{cl}(K_i + y)$. This is, the larger the size of this jump, the more we constrain the set of letters we can feasibly select in future rounds to insert into the solution. So, we divide the weight by $(\check{\rho}/K_i)(y)$ to penalize letters which induce large jumps in the closure structure. Then, in line 10, we select the letter y^* with the largest marginal adjusted weight, and save the index i^* with $y^* \in \Psi_{i^*}$. We then insert y^* into α between the letters x_{i^*} and x_{i^*+1} (or place onto the beginning or end of α when i equals 0 or ℓ respectively), and repeat until $|\alpha|$ equals the length of a basic word in $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$.

For the analysis, let α_{OPT} be an optimal word. We need to “charge” each letter in the optimal word to a decision made by the algorithm. So, first let y_t^* be the letter selected in round t and K_t^* be K_{i^*} from that round. Also, let $R = \check{\rho}(\Sigma)$. By optimism, for all letters $z \in \Sigma$, there exists $t \in \{1, \dots, R\}$ such that $z \in \text{cl}(K_t^* + y_t^*) \setminus K_t^*$. Then, let L_t be the letters from $\tilde{\alpha}_{\text{OPT}}$ for which t is the last round this occurs. Formally, this is given by the recursion over $t \in \{1, \dots, R\}$,

$$L_t \triangleq (\text{cl}(K_t^* + y_t^*) \setminus K_t^*) \setminus \left(\bigcup_{t'=t+1}^R L_{t'} \right), \quad \text{with } L_R = \text{cl}(K_R^* + y_R^*) \setminus K_R^*. \quad (8)$$

And, by our prior observation we see $(L_t)_t$ is a partition of $\tilde{\alpha}_{\text{OPT}}$. This sets up the sum,

$$\sum_{y \in \tilde{\alpha}_{\text{OPT}}} w(y) = \sum_{t=1}^R \left(\sum_{y \in L_t} w(y) \right). \quad (9)$$

But, letting $\delta_t = (\check{\rho}/K_t^*)(y_t^*)$, the choice rule makes $w(y) \leq \delta_t^{-1} w(y_t^*) \delta_t \cdot (\check{\rho}/K_{i^*})(y)$ for all $y \in L_t$, while $L_t \subseteq \text{cl}(K_t^* + y_t^*) \setminus K_t^*$ makes $(\check{\rho}/K_t^*)(y) \leq (\check{\rho}/K_t^*)(y_t^*) = \delta_t$ for all $y \in L_t$. Therefore,

$$\sum_{t=1}^R \left(\sum_{y \in L_t} w(y) \right) \leq \sum_{t=1}^R \left(|L_t| \cdot \frac{w(y_t^*)}{\delta_t} \cdot (\check{\rho}/K_{i^*})(y) \right) \leq \sum_{t=1}^R |L_t| \cdot w(y_t^*). \quad (10)$$

So, noting that $\sum_{t=1}^R w(y_t^*)$ is the value of the solution returned by Algorithm 3, we need only obtain a uniform bound on $|L_t|$ for an approximation guarantee. This is our approach. Recall from Section 1 that $\lambda_{\max}$ is the maximum value of the matroid connectivity function of the matroid defining the MPS (see Eq. 2). A careful analysis will show $|L_t| \leq 1 + \lambda_{\max}$.

► **Theorem 14.** *Let $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ be a matroidal prerequisite system, $w : \Sigma \rightarrow \mathbb{R}_+$ a nonnegative weight function, and $\text{OPT} = \max_{\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)} \sum_{y \in \alpha} w(y)$. Algorithm 3 outputs $\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$ with $\sum_{y \in \alpha} w(y) \geq (1 + \lambda_{\max})^{-1} \cdot \text{OPT}$.*

Proof. Firstly, see that α is feasible because lines 8 and 9 in Algorithm 3 ensure the following invariant: the evaluations of cl on each prefix of the working solution forms a strictly increasing chain in $(\kappa(\Lambda)/\sim)$. Since $\sigma_\rho^\vee = \text{cl}$, it can be seen then that the working solution satisfies the premise of Lemma 11 at all times. Hence, once it achieves the length of a basic word, it must be feasible.

Now, before continuing, observe the letters of the solution α are not arranged in the order in which they were inserted. In light of this, let $\pi : \{1, \dots, R\} \rightarrow \{1, \dots, R\}$ map the index corresponding to the position of a letter in α to the round index it was added to the solution. Then, we may write $\alpha = y_{\pi(1)}^* \dots y_{\pi(R)}^*$ while still identifying the round a letter was added. Furthermore, also define $X_i \triangleq \{y_{\pi(1)}^*, \dots, y_{\pi(i)}^*\}$ as the first i letters in the algorithm's solution α . Take note that as $y_{\pi(1)}^*, \dots, y_{\pi(0)}^* = \epsilon$ by convention, it follows that $X_0 = \emptyset$. Moreover, the insertion scheme of Algorithm 3 maintains,

$$\text{cl}(X_i) = \text{cl}\left(K_{\pi(i)}^* + y_{\pi(i)}^*\right), \quad \forall i \in \{1, \dots, R\}. \quad (11)$$

To see why, fix $i \in \{1, \dots, R\}$ and select an index j minimizing $\pi(j)$ among $j < i$ making $\pi(j) > \pi(i)$; this is, $y_{\pi(j)}^*$ is the first letter preceding $y_{\pi(i)}^*$ in position in α which was selected for insertion in a later round than $y_{\pi(i)}^*$ was selected. Note that if no such j exists, then Eq. 11 holds as $K_{\pi(i)}^* = \text{cl}(X_{i-1})$ would have to be true. Then, $y_{\pi(j)}^*$ belongs to a set in the sequence $(\Psi_{i'})_{i'}$ computed in line 8, and because $y_{\pi(j)}^*$ precedes $y_{\pi(i)}^*$ in position in α and minimizes $\pi(j)$ amongst such letters, this set from $(\Psi_{i'})_{i'}$ must be a subset of,

$$\text{cl}\left(\left\{y_{\pi(k)}^* \mid \pi(k) \leq \pi(i) \text{ and } k \leq i\right\}\right) = \text{cl}\left(K_{\pi(i)}^* + y_{\pi(i)}^*\right). \quad (12)$$

So, $y_{\pi(j)}^* \in \text{cl}\left(K_{\pi(i)}^* + y_{\pi(i)}^*\right)$. Now, to induct, select an arbitrary index j' with $j' < i$ and $\pi(j') > \pi(j)$. This is, $y_{\pi(j')}^*$ is any letter preceding $y_{\pi(i)}^*$ in position in α which was selected for insertion in a later round than $y_{\pi(i)}^*$ and $y_{\pi(j)}^*$. By inductive hypothesis,

$$k < i \text{ and } \pi(i) < \pi(k) < \pi(j') \implies y_{\pi(k)}^* \in \text{cl}\left(K_{\pi(i)}^* + y_{\pi(i)}^*\right).$$

Combining this with Eq. 12 shows $\text{cl}\left(\left\{y_{\pi(k)}^* \mid \pi(k) < \pi(j') \text{ and } k \leq i\right\}\right) = \text{cl}\left(K_{\pi(i)}^* + y_{\pi(i)}^*\right)$. The left hand side is the closure of the support of the prefix of the working solution ending at $y_{\pi(i)}^*$ in round $\pi(j')$. Then, because $j' < i$, i.e. $y_{\pi(j')}^*$ precedes $y_{\pi(i)}^*$ in position in α , line 8 implies $y_{\pi(j')}^* \in \text{cl}\left(K_{\pi(i)}^* + y_{\pi(i)}^*\right)$ as well. So, by the principle of induction we see $X_{i-1} \subseteq \text{cl}\left(K_{\pi(i)}^* + y_{\pi(i)}^*\right)$, which by $X_{i-1} + y_{\pi(i)}^* = X_i$ and idempotency of closure operators implies $\text{cl}(X_i) \subseteq \text{cl}\left(K_{\pi(i)}^* + y_{\pi(i)}^*\right)$. Of course, the reverse inclusion follows from,

$$\begin{aligned} \text{cl}\left(K_{\pi(i)}^* + y_{\pi(i)}^*\right) &= \text{cl}\left(\text{cl}\left(\left\{y_{\pi(k)}^* \mid k < i \text{ and } \pi(k) < \pi(i)\right\}\right) + y_{\pi(i)}^*\right), \\ &= \text{cl}\left(\left\{y_{\pi(k)}^* \mid k < i \text{ and } \pi(k) < \pi(i)\right\} + y_{\pi(i)}^*\right), \end{aligned}$$

which is a subset of $\text{cl}(X_i)$ by the monotonicity of closure operators. And so, Eq. 11 follows.

With that done, the approximation guarantee will follow by proving the following,

$$z \in L_{\pi(i)} \implies z \in \text{cl}(X_i) \setminus \text{cl}(X_{i-1}). \quad (13)$$

To see why, observe $L_{\pi(i)} \subseteq \text{cl}\left(K_{\pi(i)}^* + y_{\pi(i)}^*\right) \setminus K_{\pi(i)}^*$. So, by way of contradiction, suppose Eq. 13 does not hold. There exists $z \in L_{\pi(i)}$ with $z \notin \text{cl}(X_i) \setminus \text{cl}(X_{i-1})$. As

$z \in \text{cl}(K_{\pi(i)}^* + y_{\pi(i)}^*) \setminus K_{\pi(i)}^*$ and $\text{cl}(K_{\pi(i)}^* + y_{\pi(i)}^*) \subseteq \text{cl}(X_i)$, since $K_{\pi(i)}^* + y_{\pi(i)}^*$ is the union of the cl-closure of a subset of X_{i-1} unioned with the i^{th} letter in α , for Eq. 13 to not hold it must be that $z \in \text{cl}(X_{i-1})$. This shows,

$$(\exists z \in L_{\pi(i)}) \quad z \in \text{cl}(X_{i-1}) \setminus K_{\pi(i)}^*. \quad (14)$$

However, $\text{cl}(X_{i-1}) = \text{cl}(K_{\pi(i-1)}^* + y_{\pi(i-1)}^*)$ by Eq. 11, so by $z \in L_{\pi(i)}$ it must be that $\pi(i-1) < \pi(i)$. This means that $y_{\pi(i-1)}^*$ was in the support of the working solution when $y_{\pi(i)}^*$ was inserted. Since $y_{\pi(i-1)}^*$ directly precedes $y_{\pi(i)}^*$ in position in α , this implies $K_{\pi(i)}^* = \text{cl}(K_{\pi(i-1)}^* + y_{\pi(i-1)}^*)$. But $\text{cl}(X_{i-1}) = \text{cl}(K_{\pi(i-1)}^* + y_{\pi(i-1)}^*)$ implies $\text{cl}(X_{i-1}) = K_{\pi(i)}^*$, which contradicts Eq. 14. So, we find Eq. 13 holds true.

As a result, we observe $L_t \subseteq \tilde{\alpha}_{\text{OPT}} \cap (\text{cl}(X_i) \setminus \text{cl}(X_{i-1}))$. Now, we make use of the matroid structure to introduce $\lambda_{\max}$. Because $\tilde{\alpha}_{\text{OPT}}$ is a basic set in $\mathcal{M}$ (see Section 4),

$$|\tilde{\alpha}_{\text{OPT}} \cap \text{cl}(X_{i-1})| = r_{\mathcal{M}}(\Sigma) - r_{\mathcal{M}}(\tilde{\alpha}_{\text{OPT}} \setminus \text{cl}(X_{i-1})). \quad (15)$$

Continuing on, note that because $\text{cl}(X_i)$ is the smallest flat support containing $y_1^* \dots y_i^*$, it follows that $\text{cl}(X_i) = \kappa[y_1^* \dots y_i^*]$. This logic also shows $\text{cl}(X_{i-1}) = \kappa[y_1^* \dots y_{i-1}^*]$. Then, $\text{cl}(X_i) = \sigma_{\mathcal{M}}(\{y_1^*, \dots, y_i^*\})$ and $\text{cl}(X_{i-1}) = \sigma_{\mathcal{M}}(\{y_1^*, \dots, y_{i-1}^*\})$ by Lemma 7.1. Because feasible words in the MPS are independent in $\mathcal{M}$, we have $r_{\mathcal{M}}(\text{cl}(X_i)) = r_{\mathcal{M}}(\text{cl}(X_{i-1})) + 1$. Since independence implies $|\tilde{\alpha}_{\text{OPT}} \cap \text{cl}(X_i)| \leq r_{\mathcal{M}}(\text{cl}(X_i))$, we see $|\tilde{\alpha}_{\text{OPT}} \cap \text{cl}(X_i)| \leq r_{\mathcal{M}}(\text{cl}(X_{i-1})) + 1$. This observation combines with Eq. 15 for,

$$\begin{aligned} |L_t| &\leq |\tilde{\alpha}_{\text{OPT}} \cap (\text{cl}(X_i) \setminus \text{cl}(X_{i-1}))|, \\ &= |\tilde{\alpha}_{\text{OPT}} \cap \text{cl}(X_i)| - |\tilde{\alpha}_{\text{OPT}} \cap \text{cl}(X_{i-1})|, \\ &\leq 1 + r_{\mathcal{M}}(\text{cl}(X_{i-1})) + r_{\mathcal{M}}(\tilde{\alpha}_{\text{OPT}} \setminus \text{cl}(X_{i-1})) - r_{\mathcal{M}}(\Sigma), \\ &\leq 1 + (r_{\mathcal{M}}(\text{cl}(X_{i-1})) + r_{\mathcal{M}}(\Sigma \setminus \text{cl}(X_{i-1})) - r_{\mathcal{M}}(\Sigma)), \end{aligned}$$

which is less than or equal to $1 + \lambda_{\max}$ by definition (see Eq. 2). And so, by Eq.s 9 and 10 it is true that $\text{OPT} \leq \sum_{t=1}^R |L_t| \cdot w(y_t^*) \leq (1 + \lambda_{\max}) \cdot \sum_{t=1}^R w(y_t^*)$, which shows α is a $(1 + \lambda_{\max})$ -approximation of the optimal solution. $\blacktriangleleft$

6.2 Monotone Submodular Maximization

We can extend this method to submodular maximization by modifying the selection heuristic: the only change from Algorithm 3 is that we compute the adjusted weight of a letter y in line 10 using the marginal return $(f/\tilde{\alpha})(y)$ under f against the current working solution instead of the weight $w(y)$. We reuse the previous analysis to obtain the following.

► **Theorem 15.** *Let $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ be a matroidal prerequisite system, $f : 2^{\Sigma} \rightarrow \mathbb{R}$ a (normalized) monotone submodular function, and $\text{OPT} = \max_{\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)} f(\tilde{\alpha})$. After modifying line 10 of Algorithm 3 to compute the marginal adjusted weight $(f/\tilde{\alpha})(y)/(\check{\rho}/K_i)(y)$, the algorithm outputs α with $f(\tilde{\alpha}) \geq (2 + \lambda_{\max})^{-1} \cdot \text{OPT}$.*

Proof. Our analysis reuses parts of that presented for Algorithm 3. Notably, α is feasible by way of Lemma 11. Continuing on, define L_t in the same way as Eq. 8. Notice that the argument bounding $|L_t|$ presented in the proof of Theorem 14 only uses properties of the MPS and not the additive objective function, so we have $|L_t| \leq 1 + \lambda_{\max}$. Also let y_t^* be the

letter selected in round t , α_{OPT} and K_t^* defined as before, as well as $\alpha^{(t)}$ be the value of the working solution α at the end of round t (with the convention $\alpha^{(0)} = \epsilon$). Now, our choice rule makes,

$$\left(f/\tilde{\alpha}^{(t-1)}\right)(z) \leq \frac{\left(f/\tilde{\alpha}^{(t-1)}\right)(y_t^*)}{(\check{\rho}/K_t^*)(z)} (\check{\rho}/K_t^*)(y_t^*) \leq \left(f/\tilde{\alpha}^{(t-1)}\right)(y_t^*), \quad \forall z \in L_t,$$

and so we find that for all $t \in \{1, \dots, R\}$,

$$\sum_{z \in L_t} \left(f/\tilde{\alpha}^{(t-1)}\right)(z) \leq |L_t| \cdot \left(f/\tilde{\alpha}^{(t-1)}\right)(y_t^*) \leq (1 + \lambda_{\max}) \cdot \left(f/\tilde{\alpha}^{(t-1)}\right)(y_t^*).$$

So, because $f(\tilde{\alpha}) = \sum_{t=1}^R \left(f/\tilde{\alpha}^{(t-1)}\right)(y_t^*)$, summing over all rounds shows,

$$\sum_{t=1}^R \sum_{z \in L_t} \left(f/\tilde{\alpha}^{(t-1)}\right)(z) \leq (1 + \lambda_{\max}) \cdot f(\tilde{\alpha}). \quad (16)$$

We relate the left-hand side to OPT via $\text{OPT} \leq f(\tilde{\alpha} \cup \tilde{\alpha}_{\text{OPT}}) \leq f(\tilde{\alpha}) + \sum_{z \in \tilde{\alpha}_{\text{OPT}} \setminus \tilde{\alpha}} (f/\tilde{\alpha})(z)$, as the first inequality follows by monotonicity, while the second follows by the law of diminishing returns. Well, by the law of diminishing returns we also have $z \in L_t$ implies $(f/\tilde{\alpha}^{(t-1)})(z) \geq (f/\tilde{\alpha})(z)$ since $\tilde{\alpha}^{(t-1)} \subseteq \tilde{\alpha}$ always holds by our construction. This and Eq. 16 implies $\text{OPT} \leq f(\tilde{\alpha}) + \sum_{t=1}^R \sum_{z \in L_t} (f/\tilde{\alpha}^{(t-1)})(z) \leq (2 + \lambda_{\max}) \cdot f(\tilde{\alpha})$. Rearranging terms gives the desired result. $\blacktriangleleft$

7 Hardness of Approximation of Additive Maximization

Finally, we show that a $\min\{\Delta, \lambda_{\max}\}^{o(1)}$ -approximation for nonnegative additive maximization over an MPS is intractable. We reduce from *Densest k -Subgraph (Dk-S)*: Given an undirected graph (V, E) and integer k , select a vertex subset $S \subseteq V$ of size $|S| = k$ maximizing the induced edges. The Gap Exponential Time Hypothesis (Gap-ETH) conjectures the existence of a small $\delta > 0$ for which no subexponential time algorithm can distinguish between a satisfiable 3-SAT formula and one where at most a $(1 - \delta)$ -fraction of clauses can be satisfied [30, 9, 6]. Under the Gap-ETH, there is no $|V|^{o(1)}$ -approximation to Dk-S [29].

For a coming arithmetic exercise, define $B = |E| + 1$. We construct an MPS whose nonnegative additive maximization solutions encode Dk-S solutions in a way preserving their approximation quality. For each $u \in V$ make B “dummy” letters $d_{u,1}, \dots, d_{u,B}$, include each vertex u , and include each edge $e = \{u, v\} \in E$. For the weights set,

$$w(y) = \begin{cases} 1, & \text{if } y \text{ is an edge letter,} \\ 0, & \text{otherwise,} \end{cases} \quad (17)$$

and let the matroid $\mathcal{M}$ be the uniform matroid of rank $R \triangleq k(B + 1) + |E| + 1$. Finally, for the prerequisite order, each vertex u is greater than all its dummy letters $d_{u,1}, \dots, d_{u,B}$ and each edge $e = \{u, v\} \in E$ is greater than u and v . Recall, we showed in Section 4 that uniform matroids always satisfy compatibility with $(\Sigma, \rightsquigarrow)$. So, $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$ is a MPS.

We describe how the solutions values of the MPS and Dk-S instances are related in the following two claims. This is then used to prove Lemma 18, which gives an approximation-preserving reduction from Dk-S to nonnegative additive maximization over an MPS.

$\triangleright$ **Claim 16.** There is an efficient procedure to extract a k -subgraph with at least $\text{VAL} - 1$ edges from every basic word of cumulative weight VAL .

Proof. Let $\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$ be a basic word of value VAL, and let α' be the prefix excluding the last letter. We exclude the last letter because it can correspond to an edge which was added without both its incident vertices being present in the word. However, because the matroid is uniform, it is not possible for the prerequisites of any edge in $\tilde{\alpha}'$ to be satisfied “virtually” without their membership in the support. Therefore, because the edges correspond to the only letters with nonzero weight, if we let the cumulative weight of the support of α' be VAL' then the vertices in $\tilde{\alpha}'$ must induce a subgraph with at least VAL' edges.

Then, examine the set $X = \{v \in V \mid (\exists e \in \tilde{\alpha}' \cap E) v \in e\}$. We claim $|X| \leq k$. By way of contradiction, suppose this is not true. Our prior observations make $X \subseteq \tilde{\alpha}'$. Note then that for every letter in X , the B dummy prerequisite letters must also be present. So, if $|X| > k + 1$ it follows that there are at least $k + 1$ vertex letters and $B(k + 1)$ dummy letters in $\tilde{\alpha}'$. By recalling $B = |E| + 1$, one observes $|\tilde{\alpha}'| \geq (k + 1)(B + 1) = k(B + 1) + |E| + 2 > R$. This means $\tilde{\alpha}'$ is not independent, which contradicts the fact that it is the prefix of a basic word of $\mathcal{L}(\mathcal{M}, \rightsquigarrow)$. And so, because $|X| \leq k$, one can output a feasible Dk -S solution of value at least VAL' by listing the vertices in X along with a choice of $k - |X|$ additional vertices. Since $\text{VAL}' \geq \text{VAL} - 1$, this shows the claim. $\triangleleft$

$\triangleright$ **Claim 17.** For any Dk -S solution inducing VAL edges, there exists a basic word of cumulative weight at least VAL.

Proof. Let X be a Dk -S solution of value VAL. Take $S = \{(u, v) \in E \mid \{u, v\} \subseteq X\}$ as the edges in the induced subgraph; clearly $|S| = \text{VAL}$. Now, examine a word α formed by taking any linear extension of the ideal generated by S in $(\Sigma, \rightsquigarrow)$. Observe, $|X| = k$ implies there can be at most k vertex letters in this ideal, and each vertex letter is above exactly B letters associated with its dummies. Thus $|\alpha| \leq k(B + 1) + (\text{VAL}) \leq k(B + 1) + |E|$, and so $\tilde{\alpha}$ is independent. Furthermore, the prerequisite conditions are trivially satisfied by α since it is a linear extension of an ideal (i.e. for any $y \in \tilde{\alpha}$, every $x \rightsquigarrow y$ is such that $x \in \tilde{\alpha}$ in a position occurring before the appearance of y), so $\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)$. Since the support of α contains VAL edges, its cumulative weight is at least VAL. Therefore, any basic word formed by concatenating α with a suffix from $\mathcal{L}(\mathcal{M}, \rightsquigarrow)/[\alpha]$ gives the result. $\triangleleft$

$\blacktriangleright$ **Lemma 18.** Fix a Dk -S instance. Let Σ , $\mathcal{M}$, and $(\Sigma, \rightsquigarrow)$ as constructed, and $w : \Sigma \rightarrow \mathbb{R}$ be as defined in Eq. 17. Let OPT_{Dk-S} be the number of edges in an optimal Dk -S solution, and $\text{OPT}_{MPS} = \max_{\alpha \in \mathcal{L}(\mathcal{M}, \rightsquigarrow)} \sum_{y \in \tilde{\alpha}} w(y)$. If there exists an efficient δ^{-1} -approximation algorithm to nonnegative additive maximization over the MPS, then there exists an efficient algorithm giving a Dk -S solution inducing $\delta \cdot (\text{OPT}_{Dk-S} + O(1)) - 1$ edges.

Proof. Let OPT_{MPS} be the weight of an optimal basic word. The first claim we proved shows $\text{OPT}_{MPS} \leq \text{OPT}_{Dk-S} + 1$ while the second shows $\text{OPT}_{Dk-S} \leq \text{OPT}_{MPS}$. So, $\text{OPT}_{MPS} = \text{OPT}_{Dk-S} + O(1)$. Moreover, the first claim also shows an efficient procedure for extracting a Dk -S solution of value $\delta \cdot \text{OPT}_{MPS} - 1$ from any δ -approximation to the nonnegative additive maximization over the MPS. Altogether, $\delta \cdot \text{OPT}_{MPS} - 1 = \delta \cdot (\text{OPT}_{Dk-S} + O(1)) - 1$. $\blacktriangleleft$

Proof of Main Result 2. Now, observe the principal ideal of any edge letter is a maximum sized such ideal with cardinality exactly $2 \cdot (B + 1) + 1$. Since $R \geq 2 \cdot (B + 1)$ for $k \geq 1$, this means for any choice of edge e we have $\Delta = r_{\mathcal{M}}(\downarrow e) = \Theta(|E|) = O(|V|^2)$. Therefore, by Lemma 18, any $\Delta^{o(1)}$ -approximation algorithm for an MPS computes a $|V|^{o(1)}$ -approximation to Dk -S. Moreover, one can observe that $\lambda_{\max} \leq r_{\mathcal{M}}(\Sigma)$ is always true. And so, $\lambda_{\max} \leq R = O(|E|) = O(|V|^2)$. Thus, Lemma 18 also implies any $\lambda_{\max}^{o(1)}$ -approximation algorithm for an MPS is a $|V|^{o(1)}$ -approximation to Dk -S. $\blacktriangleleft$

References

- 1 Caterina Bigoni, Zhenying Zhang, and Jan S Hesthaven. Systematic sensor placement for structural anomaly detection in the absence of damaged states. *Computer Methods in Applied Mechanics and Engineering*, 371:113315, 2020.
- 2 Anders Björner and Günter M Ziegler. Introduction to greedoids. *Matroid applications*, 40:284–357, 1992.
- 3 Joseph E Bonin, Carolyn Chun, and Tara Fife. The natural matroid of an integer polymatroid. *SIAM Journal on Discrete Mathematics*, 37(3):1751–1770, 2023. doi:10.1137/22M1521122.
- 4 Niv Buchbinder and Moran Feldman. Deterministic algorithm and faster algorithm for submodular maximization subject to a matroid constraint. *SIAM Journal on Computing*, pages FOCS24–85, 2025.
- 5 Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011. doi:10.1137/080733991.
- 6 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-eth to fpt-inapproximability: Clique, dominating set, and more. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 743–754. IEEE, 2017. doi:10.1109/FOCS.2017.74.
- 7 Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 575–584. IEEE, 2010. doi:10.1109/FOCS.2010.60.
- 8 Asaf Cohen, Shlomi Dolev, and Guy Leshem. Sensor networks: From dependence analysis via matroid bases to online synthesis. *Theoretical Computer Science*, 553:2–17, 2014. doi:10.1016/J.TCS.2014.07.014.
- 9 Irit Dinur. Mildly exponential reduction from gap-3sat to polynomial-gap label-cover. In *Electronic colloquium on computational complexity ECCC; research reports, surveys and books in computational complexity*, page 128, 2016.
- 10 Jack Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1:127–136, 1971. doi:10.1007/BF01584082.
- 11 Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Optimization—Eureka, You Shrink! Papers Dedicated to Jack Edmonds 5th International Workshop Aussois, France, March 5–9, 2001 Revised Papers*, pages 11–26. Springer, 2003.
- 12 Ulrich Faigle. On ordered languages and the optimization of linear functions by greedy algorithms. *Journal of the ACM (JACM)*, 32(4):861–870, 1985. doi:10.1145/4221.4998.
- 13 Moran Feldman, Joseph Naor, Roy Schwartz, and Justin Ward. Improved approximations for k-exchange systems. In *European Symposium on Algorithms*, pages 784–798. Springer, 2011.
- 14 Yuval Filmus and Justin Ward. Monotone submodular maximization over a matroid via non-oblivious local search. *SIAM Journal on Computing*, 43(2):514–542, 2014. doi:10.1137/130920277.
- 15 Roy H Goetschel Jr. Linear objective functions on certain classes of greedoids. *Discrete applied mathematics*, 14(1):11–16, 1986. doi:10.1016/0166-218X(86)90003-X.
- 16 Thorkell Helgason. Aspects of the theory of hypermatroids. In *Hypergraph Seminar*, pages 191–213. Springer, 1974.
- 17 Paul Helman, Bernard ME Moret, and Henry D Shapiro. An exact characterization of greedy structures. *SIAM Journal on Discrete Mathematics*, 6(2):274–283, 1993. doi:10.1137/0406021.
- 18 Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM (JACM)*, 48(4):761–777, 2001. doi:10.1145/502090.502096.
- 19 David S Johnson and KA Niemi. On knapsacks, partitions, and a new dynamic programming technique for trees. *Mathematics of Operations Research*, 8(1):1–14, 1983. doi:10.1287/MOOR.8.1.1.

- 20 Stavros G Kolliopoulos and George Steiner. Partially ordered knapsack and applications to scheduling. *Discrete Applied Mathematics*, 155(8):889–897, 2007. doi:10.1016/J.DAM.2006.08.006.
- 21 Bernhard Korte and Dirk Hausmann. An analysis of the greedy heuristic for independence systems. In *Annals of Discrete Mathematics*, volume 2, pages 65–74. Elsevier, 1978.
- 22 Bernhard Korte and László Lovász. Greedoids—a structural framework for the greedy algorithm. In *Progress in combinatorial optimization*, pages 221–243. Elsevier, 1984.
- 23 Bernhard Korte and László Lovász. Greedoids and linear objective functions. *SIAM Journal on Algebraic Discrete Methods*, 5(2):229–238, 1984.
- 24 Bernhard Korte and László Lovász. Polymatroid greedoids. *Journal of Combinatorial Theory, Series B*, 38(1):41–72, 1985. doi:10.1016/0095-8956(85)90091-7.
- 25 Bernhard Korte, László Lovász, and Rainer Schrader. *Greedoids*, volume 4. Springer Science & Business Media, 2012.
- 26 Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2), 2008.
- 27 Jon Lee, Maxim Sviridenko, and Jan Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. *Mathematics of Operations Research*, 35(4):795–806, 2010. doi:10.1287/MOOR.1100.0463.
- 28 László Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art: Bonn 1982*, pages 235–257. Springer, 1983. doi:10.1007/978-3-642-68874-4_10.
- 29 Pasin Manurangsi. Almost-polynomial ratio hardness of approximating densest k-subgraph. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–961, 2017. doi:10.1145/3055399.3055412.
- 30 Pasin Manurangsi and Prasad Raghavendra. A Birthday Repetition Theorem and Complexity of Approximating Dense CSPs. In *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 78:1–78:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.78.
- 31 S Thomas McCormick, Britta Peis, José Verschae, and Andreas Wierz. Primal–dual algorithms for precedence constrained covering problems. *Algorithmica*, 78(3):771–787, 2017. doi:10.1007/S00453-016-0174-3.
- 32 Julián Mestre. Greedy in approximation algorithms. In *European Symposium on Algorithms*, pages 528–539. Springer, 2006. doi:10.1007/11841036_48.
- 33 Rolf H. Möhring, Martin Skutella, and Frederik Stork. Scheduling with and/or precedence constraints. *SIAM Journal on Computing*, 33(2):393–415, 2004. doi:10.1137/S009753970037727X.
- 34 George L Nemhauser and Laurence A Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978. doi:10.1287/MOOR.3.3.177.
- 35 George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978. doi:10.1007/BF01588971.
- 36 James B Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009. doi:10.1007/S10107-007-0189-2.
- 37 James G Oxley. *Matroid theory*, volume 3. Oxford University Press, USA, 2006.
- 38 Jean-Claude Picard. Maximal closure of a graph and applications to combinatorial problems. *Management science*, 22(11):1268–1272, 1976.
- 39 Thomas Powers, David W Krout, Jeff Bilmes, and Les Atlas. Constrained robust submodular sensor selection with application to multistatic sonar arrays. *IET Radar, Sonar & Navigation*, 11(12):1776–1781, 2017.

- 40 Richard Rado. Note on independence functions. *Proceedings of the London Mathematical Society*, 3(1):300–320, 1957.
- 41 Natthawut Samphaiboon and Y Yamada. Heuristic and exact algorithms for the precedence-constrained knapsack problem. *Journal of optimization theory and applications*, 105(3):659–676, 2000.
- 42 Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000. doi:10.1006/JCTB.2000.1989.
- 43 Alexander Schrijver et al. *Combinatorial optimization: polyhedra and efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.
- 44 Robert P Streit and Vijay K Garg. The polymatroid representation of a greedoid, and associated galois connections. *arXiv preprint*, 2026. arXiv:2411.15363.
- 45 Maxim Sviridenko, Jan Vondrák, and Justin Ward. Optimal approximation for submodular and supermodular optimization with bounded curvature. *Mathematics of Operations Research*, 42(4):1197–1218, 2017. doi:10.1287/MOOR.2016.0842.
- 46 Dávid Szeszlér. Sufficient conditions for the optimality of the greedy algorithm in greedoids. *Journal of Combinatorial Optimization*, 44(1):287–302, 2022. doi:10.1007/S10878-021-00833-Y.
- 47 William T Tutte. Connectivity in matroids. *Canadian Journal of Mathematics*, 18:1301–1324, 1966.
- 48 Dominic JA Welsh. *Matroid theory*. Courier Corporation, 2010.