Combinatorial Algorithms for Submodular Function Minimization and Related Problems

by

Christopher Price

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2015

Abstract

Submodular functions are common in combinatorics; examples include the cut capacity function of a graph and the rank function of a matroid. The submodular function minimization problem generalizes the classical minimum cut problem and also contains a number of other combinatorial optimization problems as special cases. In this thesis, we study submodular function minimization and two related problems: matroid polyhedron membership and matroid intersection. A significant contribution concerns algorithms for the latter problems due to Cunningham. In particular, we identify and correct errors in the original proofs of the running time bounds for these algorithms.

Acknowledgements

I would like to thank Bill Cunningham for his supervision. Without his guidance, patience and humility, this work would not have been possible. Additionally, I would like to thank my readers Joseph Cheriyan, Bertrand Guenin and Chaitanya Swamy for their valuable input and suggestions. Finally, I would like to thank NSERC, OGS and the University of Waterloo for their generous financial support.

Table of Contents

List of Figures

List of Algorithms and Subroutines

Chapter 1

Preliminaries

1.1 Introduction

In 1984, Cunningham [3] gave the first efficient combinatorial algorithm for the matroid polyhedron membership problem. The algorithm, like classical network flow algorithms, iteratively constructs an auxiliary digraph and performs augmentations along a shortest source-sink path. The running time bound depends on a monotonicity property of path lengths in the auxiliary digraph.

Cunningham also applied these ideas to improve the running time of the matroid intersection algorithm [5]. Once again, the algorithm iteratively constructs an auxiliary digraph and performs augmentations along shortest source-sink paths. Furthermore, the argument for the running time bound depends on a similar path-length monotonicity property.

The approach taken by Cunningham to solving the matroid polyhedron membership problem was adapted by Schrijver [21] and, independently, by Iwata, Fleischer and Fujishige [16], to give efficient combinatorial algorithms for submodular function minimization, solving a problem that remained open for many years. Moreover, the running time of Cunningham's improved version of the matroid intersection algorithm is still the best known. While the main results of [3] and [5] are correct, their original proofs are not. In particular, the path-length monotonicity properties on which the running time bounds depend are not true in general. However, in both cases, a weakened form of the monotonicity property can be proven, and can in turn be used to provide complete and correct proofs of the main results.

In the case of matroid intersection, the errors in [3] have been previously considered. In his German language diploma thesis, Haselmayr [14] gives a counterexample to the path-length monotonicity property, a proof of its weakened form, and a proof of the running time of the algorithm. In the case of matroid polyhedron membership, analogous results and proofs appear in this thesis for the first time.

A primary purpose of this thesis is to study and correct the errors in Cunningham's papers. Most significantly, we establish the validity of Cunningham's algorithm for matroid polyhedron membership. Additionally, we consider Haselmayr's work [14] on the matroid intersection algorithm, and offer a similar resolution to the problem. Finally, we study the problem of submodular function minimization, with an emphasis on its relationship to the matroid polyhedron membership problem.

The remainder of Chapter 1 introduces some background in graph theory and matroid theory.

In Chapter 2, we study the matroid polyhedron membership problem. We begin by presenting Cunningham's algorithm. Next, we identify and examine the error found in [3]. After correcting the error, we prove a modified monotonicity property and go on to prove that the algorithm is indeed polynomial-time. In the last section of Chapter 2, we consider a slight variant of the algorithm for which Cunningham's original argument is valid.

Chapter 3 studies the matroid intersection problem. In particular, we present Cunningham's algorithm and a counterexample to the monotonicity property of [5]. In addition, we study the work of Haselmayr, and offer a proof of the polynomial running time bound originally claimed.

Chapter 4 studies submodular function minimization, which contains both matroid polyhedron membership and the decision version of matroid intersection as special cases. We begin by presenting Schrijver's algorithm [21]. By studying the behaviour of the algorithm in the special case of matroid polyhedron membership, we arrive at another matroid polyhedron membership algorithm. Next, we consider the work of Frank and Miklós [12], who give a push-relabel algorithm for matroid polyhedron membership. Finally, we extend the algorithm of [12] to obtain a push-relabel algorithm for submodular function minimization, due to Fleischer and Iwata [10].

1.2 Graph Theory

In what follows, we require some basic terminology from graph theory. Standard terms and symbols are taken from the text *Graph Theory* by Bondy and Murty [2].

Let $D = (N, A)$ be a digraph. For distinct nonadjacent vertices $s, t \in N$, a set $N' \subseteq N \setminus \{s, t\}$ is called an *st-separator* if there is no st-dipath in the subdigraph of D induced by $N \setminus N'$. Two st-dipaths are said to be *disjoint* if their only common vertices are s and t. We will require the following theorem of Menger [2].

Theorem 1.2.1. (Menger's Theorem) *Let $D = (N, A)$ be a digraph and let $s, t \in N$ be two distinct vertices such that $(s, t) \notin A$. Then the minimum cardinality of an st-separator is equal to the maximum number of pairwise disjoint st-dipaths.*

The notation $d(s, u)$ denotes the length of a shortest su-dipath in D, and where $X, Y \subseteq N$, $d(X, Y)$ denotes the length of a shortest dipath from a vertex in X to a vertex in Y. We call an e_1e_k-dipath e_1,e_2, ...,e_k in D *chordless* if there is no arc e_ie_j for any pair i, j satisfying $1 < i + 1 < j \leq k$.

We will also need to find a shortest path in a digraph. For this, we use a common technique, known as *breadth-first search* . Breadth-first search accepts a digraph (N, A) and a distinguished vertex s as inputs and terminates in $O(|N|^2)$ time. It works by visiting each vertex reachable from s precisely once and considering its neighbours. It chooses the next vertex to visit by a first-in, first-out rule. Given an ordering on the vertices, if breadth-first search considers the neighbours of each vertex in increasing order, then it will determine a *lexicographically least shortest st-dipath* , that is, the unique shortest st-dipath whose vertex sequence is lexicographically least.

1.3 Matroid Theory

A *matroid* is a set system $(S,\mathcal{I})$ consisting of a *ground set* S together with a set $\mathcal{I}$ of subsets of S satisfying the following axioms.

$(M0)$ $\emptyset \in \mathcal{I}$.

$(M1)$ If $J \in \mathcal{I}$ and $J' \subseteq J$, then $J' \in \mathcal{I}$.

$(M2)$ If $J, J' \in \mathcal{I}$ and $|J'| = |J| + 1$, then there exists $e \in J' \setminus J$ such that $J + e \in \mathcal{I}$.

Throughout this thesis, the identifier n universally refers to the size of the ground set of the nearest matroid. The results in this section are all standard, and can be found in the text *Matroid Theory* by Oxley [19].

Members of $\mathcal{I}$ are called *independent sets* . Given $A \subseteq S$, we call an inclusionwise maximal independent subset $J \subseteq A$ an *M-basis* of A, or where unambiguous, simply a *basis* of A. A basic result in matroid theory states that, for any $A \subseteq S$, every basis of A is equicardinal.

Proposition 1.3.1. *Let $M = (S, \mathcal{I})$ be a matroid and $A \subseteq S$. Let J and J' be two maximal independent subsets of A. Then $|J| = |J'|$.*

Proof. Suppose $|J'| > |J|$. If $|J'| = |J| + 1$, then by axiom (M2), J is not a maximal independent subset of A. If $|J'| > |J| + 1$, then by (M2), for any subset I of J' with $|I| = |J| + 1$, there exists some $e \in I \setminus J$ such that $J + e \in \mathcal{I}$, again contradicting the maximality of J. $\square$

We refer to the cardinality of a basis of A as the *rank* of A. Associated with any matroid $M = (S, \mathcal{I})$ is its *rank function* $r_M : 2^S \to \mathbb{Z}$ which maps each $A \subseteq S$ to its rank. Where unambiguous, we abbreviate r_M to r. The following *submodular* property of the rank function will be useful in later chapters.

Lemma 1.3.2. *Let $M = (S, \mathcal{I})$ be a matroid and let $A, B \subseteq S$. Then*

$$r(A) + r(B) \geq r(A \cup B) + r(A \cap B).$$

Proof. Let J_1 be a basis of $A \cap B$, and extend J_1 to a basis J_2 of $A \cup B$. Then $A \cap J_2 \in \mathcal{I}$ and $B \cap J_2 \in \mathcal{I}$, so $r(A) \geq |J_2 \cap A|$ and $r(B) \geq |J_2 \cap B|$. Now,

$$\begin{aligned} r(A) + r(B) &\geq |A \cap J_2| + |B \cap J_2| \\ &= |J_2 \cap (A \cap B)| + |J_2 \cap (A \cup B)| \\ &= |J_1| + |J_2| \\ &= r(A \cap B) + r(A \cup B). \end{aligned}$$

$\square$

A set $A \subseteq S$ is called *dependent* if is it not independent. A *circuit* is an inclusionwise minimal dependent subset of S. For an independent set J and an element $e \in S \setminus J$, if $J + e \notin \mathcal{I}$, then $J + e$ must contain a circuit. The following result says that there is a unique circuit in $J + e$.

Lemma 1.3.3. *Let $M = (S, \mathcal{I})$ be a matroid, let $J \in \mathcal{I}$, and let $e \in S$ such that $J + e \notin \mathcal{I}$. Then $J + e$ contains precisely one circuit C and $e \in C$.*

Proof. Suppose not. Choose $J \in \mathcal{I}$, $e \in S \setminus J$ such that $J + e$ contains two circuits C_1 and C_2. Subject to this, choose J as small as possible. Then $J = (C_1 \cup C_2) - e$. As $C_1 \neq C_2$, there exists $a \in C_1 \setminus C_2$ and $b \in C_2 \setminus C_1$. It follows that $(C_1 \cup C_2) \setminus \{a, b\}$ contains a basis B of $C_1 \cup C_2$. But J is a basis of $C_1 \cup C_2$ and $|B| \leq |(C_1 \cup C_2) \setminus \{a, b\}| < |J|$, a contradiction. □

We denote this unique circuit, if it exists, by $C_M(J, e)$, and as always, we omit the subscript where unambiguous. Given matroids $M_1, M_2, ..., M_k$, if J is independent in M_i and $J + e$ is not, then we denote the circuit of M_i contained in $J + e$ by $C_i(J, e)$. The Strong Circuit Axiom is another standard result that we will use regularly.

Lemma 1.3.4. (Strong Circuit Axiom) *Let $M = (S, \mathcal{I})$ be a matroid and let C_1, C_2 be two distinct circuits of M with $e \in C_1 \cap C_2$ and $f \in C_1 \setminus C_2$. Then there exists a circuit C_3 of M with $f \in C_3 \subseteq (C_1 \cup C_2) - e$.*

Proof. By Lemma 1.3.2, $r(C_1) + r(C_2) \geq r(C_1 \cup C_2) + r(C_1 \cap C_2)$. Since $C_1 \cap C_2$ is a proper subset of C_1, $r(C_1 \cap C_2) = |C_1 \cap C_2|$. Since C_1 and C_2 are circuits, $r(C_1) = |C_1| - 1$ and $r(C_2) = |C_2| - 1$. Substituting these values into the inequality and rearranging gives $r(C_1 \cup C_2) \leq |C_1 \cup C_2| - 2$. It follows that $C_1 \cup C_2 - e \notin \mathcal{I}$, so there is a circuit $C_3 \subseteq C_1 \cup C_2 - e$.

Suppose no circuit of $C_1 \cup C_2 - e$ contains f. Subject to this, choose C_1 and C_2 so that $|C_1 \cup C_2|$ is as small as possible. Evidently $e \in C_2 \setminus C_3$ and since $C_3 \nsubseteq C_1$, there exists some $g \in C_3 \cap (C_2 \setminus C_1)$. Since $f \in (C_1 \cup C_2) \setminus (C_2 \cup C_3)$, $|C_2 \cup C_3| < |C_1 \cup C_2|$. Then by the choice of C_1 and C_2, there exists a circuit $C_4 \subseteq C_2 \cup C_3 - f$ with $e \in C_4$. Now, $e \in C_1 \cap C_4$ and $f \in C_1 \setminus C_4$. As $C_1 \cup C_4 \subseteq C_1 \cup C_2 - g$, there exists a circuit $C_5 \subseteq C_1 \cup C_4 - e \subseteq C_1 \cup C_2 - e$ and $f \in C_5$, a contradiction. □

Let $M = (S, \mathcal{I})$ be a matroid and let $A \subseteq S$. Then for any element $e \in S \setminus A$, $r(A) \leq r(A + e) \leq r(A) + 1$. If $r(A + e) = r(A)$, then e is said to be in the closure of A. The *closure* of the set A, denoted $cl_M(A)$, is equal to the set of all elements $e \in S$ such that $r(A + e) = r(A)$. In what follows, we require the following results about the closure operator.

Proposition 1.3.5. *Let $M = (S, \mathcal{I})$ be a matroid and let J be a basis of $A \subseteq S$. Then J is also a basis of $cl(A)$.*

Proof. Suppose not. Then there exists $e \in cl(A) \setminus A$ such that $J + e \in \mathcal{I}$. As $e \in cl(A)$, $r(A + e) = r(A)$. But $J + e \subseteq A + e$ and $J + e \in \mathcal{I}$, so $r(A + e) \geq |J| + 1 > r(A)$, a contradiction. $\square$

Lemma 1.3.6. *Let $M = (S, \mathcal{I})$ be a matroid with rank function r and let $A \subseteq S$. Then $r(A) = r(cl(A))$.*

Proof. This lemma is a direct corollary of Proposition 1.3.5. Since a basis J of A is a basis of $cl(A)$, $r(A) = |J| = r(cl(A))$, as required. $\square$

Lemma 1.3.7. *Let $M = (S, \mathcal{I})$ be a matroid and let J be a basis of $A \subseteq S$. Then $cl(J) = cl(A)$.*

Proof. Let $e \in cl(A)$. Then $r(A + e) = r(A)$. If $e \notin cl(J)$, then $J + e \in \mathcal{I}$. But then J is not a basis of $cl(A)$, contradicting Proposition 1.3.5.

Now let $e \in cl(J)$. Then $r(J + e) = r(J)$, so $J + e \notin \mathcal{I}$. On the other hand, if $e \notin cl(A)$, then $r(A + e) > r(A)$, which implies that $J + e \in \mathcal{I}$, a contradiction. $\square$

Lemma 1.3.8. *Let $M = (S, \mathcal{I})$ be a matroid and let $A \subseteq S$. Let $B \subseteq \mathrm{cl}(A)$. Then $\mathrm{cl}(A \cup B) = \mathrm{cl}(A)$.*

Proof. Let J be a basis of A. By Proposition 1.3.5, J is also a basis of $cl(A)$. Since $B \subseteq cl(A)$, we have $A \subseteq A \cup B \subseteq cl(A)$. Now $r(A) = r(cl(A))$ by Lemma 1.3.6. But $r(A) \leq r(A \cup B) \leq r(cl(A))$, so $r(A \cup B) = r(A)$. It follows that J is a basis of $A \cup B$. Now by Lemma 1.3.7, $cl(A) = cl(J) = cl(A \cup B)$, as required. $\square$

Arguments concerning the running time of matroid algorithms are slightly unusual since it is possible to have exponentially many independent sets in a matroid. Consequently, if we allow the collection of independent sets of a matroid as the input, then we might achieve a 'polynomial time' algorithm in the length of the input, which is itself exponential in n. One might think that this necessarily precludes the possibility of a polynomial-time algorithm for optimization problems on matroids, but this is not the case. In many concrete classes of matroids, even if there are exponentially many independent sets, there exist polynomial-time algorithms which determine if a set is independent. For a matroid $M = (S, \mathcal{I})$, we call an algorithm which, given some $A \subseteq S$, returns true if $A \in \mathcal{I}$ and false otherwise, an *independence oracle* for M. For any matroid algorithm, we assume that the input matroid is represented as a ground set together with an independence oracle.

Since independence oracles can have quite different running times, it makes sense to abstract a matroid algorithm from its independence oracle. This allows us to give a generic running time, as a function of n and the running time of the independence oracle. We call a matroid algorithm a *polynomial time matroid algorithm* if its running time is bounded from above by a function polynomial in n and the running time of the independence oracle. Note that, for a concrete implementation, a polynomial time matroid algorithm is polynomial time if and only if the independence oracle is polynomial time.

Chapter 2

Matroid Polyhedron Membership

2.1 Introduction

In this chapter, we study the matroid polyhedron membership problem. Given a matroid $M = (S, \mathcal{I})$ with rank function r, the *independent set polyhedron* of M, denoted P_M, is equal to the convex hull of (incidence vectors of) independent sets of M. In the *matroid polyhedron membership problem*, we are given a matroid $M = (S, \mathcal{I})$ and a vector $x \in \mathbb{R}^S$, and asked to determine if $x \in P_M$.

One might immediately wonder if an efficient algorithm is likely to exist for this problem, that is, are there succinct certificates which verify a 'yes' and a 'no' answer? The answer to this question is provided by the following theorem of Edmonds [7].

Theorem 2.1.1. (Matroid Polytope Theorem) *For any matroid $M = (S, \mathcal{I})$ with rank function r, the system*

$$\begin{aligned} x(A) &\leq r(A)\ \forall A \subseteq S \\ x &\geq 0 \end{aligned}$$

defines the convex hull of independent sets of M.

It follows that, if $x \in P_M$, then there exists an expression for x as a convex combination of independent sets of M. Of course, in the event that $x \notin P_M$, we would like a certificate, verifiable in polynomial time, that this is the case. Clearly, if $x \not\geq 0$, then $x_e < 0$ for some

$e \in S$, so the set $\{e\}$ is a certificate. Providing $x \geq 0$, if $x \notin P_M$, then there must exist some $A \subseteq S$ with $x(A) > r(A)$; in this case, we would like to return such a set.

The Matroid Polytope Theorem came in 1970, but no efficient algorithm for solving the matroid polyhedron membership problem was known even at the time of Khachiyan's 1979 proof that the ellipsoid method represented a polynomial-time algorithm for solving linear programs [17]. This proof generated interest in the ellipsoid method and its potential applications to combinatorial optimization. In particular, a 1981 paper by Grötschel, Lovász, and Schrijver [13] studied a number of problems through the lens of the ellipsoid method. One of the focal points of this paper was to prove the equivalence of the separation problem and the optimization problem.

Let $P \subseteq \mathbb{R}^n$ be a polyhedron. The *optimization problem* is, given a vector $w \in \mathbb{R}^n$, to find a vector $x \in P$ such that $w^\top x$ is maximized. The *separation problem* is, given a vector $x \in \mathbb{R}^n$, to determine whether or not $x \in P$, and if not, to find a hyperplane separating x from P. The theorem of Grötschel, Lovász, and Schrijver states that there is a polynomial-time algorithm to solve the separation problem over a class of polyhedra if and only if there is a polynomial-time algorithm to solve the optimization problem over that same class.

In our case, we are dealing with matroid polyhedra, that is, those described by the set of inequalities in the Matroid Polytope Theorem. Furthermore, the matroid polyhedron membership problem is merely an instance of the separation problem for this class of polyhedra. Fortunately, as Edmonds proved in 1971 [8], this is a class of polyhedra for which the greedy algorithm always delivers an optimal solution. It follows that we can solve the optimization problem over this class, and therefore we must be able to solve the separation problem also.

Consider the following theorem of Edmonds, a special case of his "Polymatroid Intersection Theorem" [7].

Theorem 2.1.2. (Membership Min-Max Theorem) *Let $M = (S, \mathcal{I})$ be a matroid and let $x \in \mathbb{R}^S$. Then*

$$\max_{y \in P_M : y \leq x} \{y(S)\} = \min_{A \subseteq S} \{r(A) + x(S \setminus A)\} \tag{2.1}$$

Evidently, $x \in P_M$ if and only if the maximizing vector in 2.1 is equal to x. Otherwise, if A minimizes $r(A) + x(S \setminus A)$, then it also minimizes $r(A) - x(A)$, that is, it provides an inequality $x(A) \leq r(A)$ which is most violated by x. The proof of Theorem 2.1.2 is not difficult, and relies on the following lemma, which also appears in [7].

Lemma 2.1.3. *Let $M = (S, \mathcal{I})$ be a matroid, let $x \in \mathbb{R}^S$ and let $y \in P_M$. If $A, B \subseteq S$ are such that $y(A) = r(A)$ and $y(B) = r(B)$, then $y(A \cup B) = r(A \cup B)$ and $y(A \cap B) = r(A \cap B)$.*

Proof. Let $A, B \subseteq S$ be such that $y(A) = r(A)$ and $y(B) = r(B)$. Then

$$\begin{aligned} y(A) + y(B) &= y(A \cup B) + y(A \cap B) \\ &\leq r(A \cup B) + r(A \cap B) && \text{(Theorem 2.1.1)} \\ &\leq r(A) + r(B) && \text{(submodularity)} \\ &= y(A) + y(B). \end{aligned}$$

Equality must hold throughout. In particular, $y(A \cup B) = r(A \cup B)$ and $y(A \cap B) = r(A \cap B)$. $\square$

Proof. (Of the Membership Min-Max Theorem (2.1.2)) Let $Z \subseteq S$ and let $y \in P_M$ with $y \leq x$. Then

$$\begin{aligned} y(S) &= y(Z) + y(S \setminus Z) \\ &\leq r(Z) + y(S \setminus Z) && (y \in P_M) \\ &\leq r(Z) + x(S \setminus Z). && (y \leq x) \end{aligned}$$

Therefore, the maximum in (2.1) is less than or equal to the minimum. It remains to show that equality always holds.

Suppose y is a maximizing vector in (2.1). Observe that, if for some $e \in S$, we have $y_e < x_e$, then e belongs to some set $Y \subseteq S$ with $y(Y) = r(Y)$. If this were not the case, then $y + \varepsilon\chi_{\{e\}} = y' \in P_M$ and $y' \leq x$ for some sufficiently small $\varepsilon > 0$, contradicting that y is a maximizer.

Define A to be the union of all sets $X \subseteq S$ with $y(X) = r(X)$. By Lemma 2.1.3, $y(A) = r(A)$. Since each $e \in S \setminus A$ has $y_e = x_e$, we also have $y(S \setminus A) = x(S \setminus A)$. Thus $y(S) = r(A) + x(S \setminus A)$, as required. $\square$

The previous discussion having established that the problem is well characterized, we now begin to describe an algorithm that will yield an optimal pair. Sections 2 through 5 of this chapter closely follow the work of Cunningham [3], who gives a combinatorial algorithm for the matroid polyhedron membership problem. Beyond understanding the problem at hand, the purpose of the chapter is to clarify the presentation of the algorithm, as well as to correct Cunningham's proof of its running time.

Section 2 of this chapter presents the basic ideas behind the algorithm, in particular the notion of a crude augmentation. Section 3 goes on to refine the crude augmentation to produce a grand augmentation. Section 4 discusses and corrects the error found in Cunningham's paper, and Section 5 proves the bound on the running time of the algorithm. Section 6 presents a similar algorithm for the matroid polyhedron membership problem.

2.2 The Crude Augmentation

Throughout this section, let $M = (S, \mathcal{I})$ and $x \in \mathbb{R}^S$ be an instance of the matroid polyhedron membership problem. The algorithm is an iterative one, and at a high level, it works as follows. We keep a vector $\lambda \in \mathbb{R}^{\mathcal{I}}$ satisfying

$$\lambda \geq 0 \text{ and } \sum_{J \in \mathcal{I}} \lambda_J = 1 \tag{2.2}$$

and for any such vector λ, define

$$y(\lambda) = \sum_{J \in \mathcal{I}} \lambda_J \chi_J,$$

where χ_J is the incidence vector of the independent set J. Where unambiguous, we write J instead of χ_J. Also where unambiguous, we refer to $y(\lambda)$ as y and $y(\lambda')$ as y'. Since y is a convex combination of independent sets, we know that $y \in P_M$.

Clearly, the vector λ may have exponentially many components if $\mathcal{I}$ has exponentially many sets. Using such a vector is simply a notational convenience; the algorithm will only consider those sets $J \in \mathcal{I}$ for which $\lambda_J > 0$. We call such a set J *active* , and we will show that the algorithm only creates polynomially many active sets.

The main loop of the algorithm begins with some λ such that $y \leq x$. If $y = x$, then the algorithm terminates and λ is a certificate that $x \in P_M$. If instead $y \neq x$, then the algorithm seeks a vector λ' such that $x \geq y' \geq y$, and $y' \neq y$. If no such vector exists, then $x \notin P_M$, and the algorithm seeks a set $A \subseteq S$ to certify this.

Suppose, for some $\lambda \in \mathbb{R}^{\mathcal{I}}$, that $y \leq x$, $x \in P_M$, and $y \neq x$. How can we find a vector λ' such that $x \geq y' \geq y$ and $y' \neq y$? In the simplest case, there is some $e \in S$ such that $y_e < x_e$ and some active J such that $J' = J + e \in \mathcal{I}$. In this case, we can set

$$\lambda'_{J'} = \lambda_{J'} + \min\{x_e - y_e, \lambda_J\} \text{ and } \lambda'_J = \lambda_J - \min\{x_e - y_e, \lambda_J\} \text{ and } \lambda'_I = \lambda_I \; \forall I \neq J$$

Then $y' = y + (\min\{x_e - y_e, \lambda_J\}) \cdot \{e\}$. Of course, this case will not always occur. It could be that, for each $e \in S$ with $y_e < x_e$, and for each active J, $J + e \notin \mathcal{I}$. In this case, we can still find a λ' satisfying the given conditions if we can find a sequence $e_1, e_2, ..., e_k$ of elements of S and a sequence $J_1, J_2, ..., J_k$ of (not necessarily distinct) active independent sets such that

$$e_i \notin J_i \text{ for all } 1 \leq i \leq k \tag{2.3}$$

$$e_{i+1} \in J_i \text{ for all } 1 \leq i < k \tag{2.4}$$

$$y_{e_1} < x_{e_1}, \tag{2.5}$$

$$J_i + e_i - e_{i+1} \in \mathcal{I} \text{ for all } 1 \leq i < k, \text{ and} \tag{2.6}$$

$$J_k + e_k \in \mathcal{I}. \tag{2.7}$$

In this case, it is not too hard to see how to find λ'; essentially, we take all the sets $J'_i = J_i + e_i - e_{i+1}$ for $i = 1, ..., k-1$ and $J'_k = J_k + e_k$ and add some $\delta > 0$ to their coefficients, and we subtract that same δ from the coefficient of each J_i. There are of course some special considerations for sets J_i which appear more than once in the sequence $J_1, J_2, ..., J_k$, which our proof of the Augmenting Path Theorem expounds upon.

At this point, we hypothesize that we can always find a sequence satisfying (2.3)-(2.7) if $x \in P_M$, $y \leq x$ and $y \neq x$. In order to find such a sequence, we construct an *auxiliary digraph* $D(M, x, \lambda) = (N, A)$ as follows. Let $N = S \cup \{s, t\}$, where $s, t \notin S$ with the following arc-set:

- se whenever $y_e < x_e$
- et whenever there exists an active J with $e \notin J$ such that $J + e \in \mathcal{I}$

- ef whenever there exists an active J with $e \notin J$, $f \in J$ such that $J + e - f \in \mathcal{I}$

The following theorem formalizes the above discussion, and also shows how to find a certificate in the event that no sequence satisfying (2.3)-(2.7) exists.

Theorem 2.2.1. (Membership Augmenting Path Theorem) *Let $\lambda \in \mathbb{R}^{\mathcal{I}}$ such that $y \leq x$ and λ satisfies (2.2). Let $D = D(M, x, \lambda)$. If there exists an st-dipath in D then there exists $\lambda' \in \mathbb{R}^{\mathcal{I}}$ satisfying (2.2) such that $y \leq y' \leq x$ and $y' \neq y$. If there is no st-dipath in D and $y \neq x$, then there exists $A \subseteq S$ such that $x(A) > r(A)$. Moreover, (y, A) is an optimal pair in the Membership Min-Max Theorem (2.1.2).*

Proof. Suppose there exists an st-dipath P in D with vertex sequence $s, e_1, e_2, ..., e_k, t$. By construction, there exist $J_1, J_2, ..., J_k$ (not necessarily distinct) satisfying (2.3)-(2.7).

For i with $1 \leq i \leq k$, let μ_i denote the number of times that J_i appears in the sequence $J_1, ..., J_k$. Consider the following subroutine.

Subroutine 1 Crude Augmentation of λ using st-dipath P

```
$\delta \leftarrow \min_{1 \leq i \leq k} \left\{ \frac{\lambda_{J_i}}{\mu_i} \right\}$
$\lambda' \leftarrow \lambda$
for $i = 1$ to $k - 1$ do
    $J'_i \leftarrow J_i + e_i - e_{i+1}$
    $\lambda'_{J'_i} \leftarrow \lambda_{J'_i} + \delta$
    $\lambda'_{J_i} \leftarrow \lambda'_{J_i} - \delta$
end for
$J'_k \leftarrow J_k + e_k$
$\lambda'_{J'_k} \leftarrow \lambda_{J'_k} + \delta$
$\lambda'_{J_k} \leftarrow \lambda'_{J_k} - \delta$
return $\lambda'$
```

It is easy to see that the above subroutine returns λ' satisfying (2.2). Consider some element e_i with $2 \leq i \leq k$. We have

$$J'_{i-1} = J_{i-1} + e_{i-1} - e_i \text{ and } J'_i = J_i + e_i - e_{i+1} \text{ or } J'_k = J_k + e_k.$$

In particular, $e_i \in J'_i$ and $e_i \in J_{i-1}$. Furthermore, we have

$$\lambda'_{J_{i-1}} = \lambda_{J_{i-1}} - \delta \text{ and } \lambda'_{J'_i} = \lambda_{J'_i} + \delta.$$

We conclude that $y_e = y'_e$ for all $e \neq e_1$. There is only one set change involving e_1, namely $J'_1 = J_1 + e_1 - e_2$. The new coefficients are

$$\lambda'_{J'_1} = \lambda_{J'_1} + \delta \text{ and } \lambda'_{J_1} = \lambda_{J_1} - \delta.$$

It follows that $y' = y + \delta \{e_1\}$.

Conversely, suppose $y \neq x$ and there exists no st-dipath in D. Let $A = \{a \in S :$ there exists an sa-dipath in $D\}$. We claim that, for each active J, $J \cap A$ is a basis for A. Suppose not. Then there exists some active J and some $e \in A \setminus J$ such that $(J \cap A) + e \in \mathcal{I}$. If $J + e \in \mathcal{I}$, then et is an arc of D, a contradiction. If $J + e \notin \mathcal{I}$, then since $(J \cap A) + e \in \mathcal{I}$, we know that $C(J, e) \nsubseteq A$. Therefore, there exists some $f \in C(J, e) \setminus (J \cap A)$. It follows from Lemma 1.3.3 that ef is an arc of D. But $e \in A$ and $f \notin A$, a contradiction.

Now,

$$\begin{aligned} y(A) &= \sum_{J \in \mathcal{I}} \lambda_J \cdot \chi_J(A) \\ &= \sum_{J \in \mathcal{I}} \lambda_J \cdot |J \cap A| \\ &= \sum_{J \in \mathcal{I}} \lambda_J \cdot r(A) \\ &= r(A), && \text{(by 2.2)} \end{aligned}$$

where the third equality follows from the fact that $J \cap A$ is a basis for A. Since $y \neq x$ and $y \leq x$, we know that $x(A) > y(A) = r(A)$. By construction, every $e \in S \setminus A$ has $y_e = x_e$. Therefore, $y(S \setminus A) = x(S \setminus A)$. Now, $y(S) = y(A) + y(S \setminus A) = r(A) + x(S \setminus A)$, so (y, A) is an optimal pair in (2.1). □

We nearly have the necessary components of an algorithm: we have a termination condition and a subroutine which ostensibly brings the algorithm closer to termination. What we cannot be certain of is the number of iterations required to achieve termination, or indeed, if the algorithm even terminates. Placing a bound on the running time of the algorithm will require some significant modifications, including a substantive change to the augmentation step. This is the subject of the following section. In addition, proving the bound will require a deep understanding of the behaviour of the digraph from iteration to iteration, which is the subject of subsequent sections.

2.3 The Grand Augmentation

For this section, let $M = (S, \mathcal{I})$ and $x \in \mathbb{R}^S$ be an instance of the matroid polyhedron membership problem. Additionally, suppose we have some $\lambda \in \mathbb{R}^{\mathcal{I}}$ satisfying (2.2) such that $y \leq x$, and let D be the auxiliary digraph with respect to x and λ. Let Q be a chordless st-dipath in D. As alluded to in the previous section, it is not apparent how to place a bound on the number of augmentations that will occur in the naïve algorithm. One of the main problems is that if we perform a crude augmentation using path Q, it is possible that Q will be an st-dipath in the next auxiliary digraph. It is not even clear how to place a bound on the number of augmentations along Q required to ensure that some arc of Q disappears. The section begins with some definitions and motivation for the grand augmentation, and concludes by defining the grand augmentation and proving some of its properties.

There may be multiple independent sets giving rise to a particular arc of Q. For each arc ef of Q with $e \neq s$, let

$$D(e,f) = \begin{cases} \{J \in \mathcal{I} : \lambda_J > 0,\ f \in C(J,e)\} & : f \neq t \\ \{J \in \mathcal{I} : \lambda_J > 0,\ e \notin J,\ J + e \in \mathcal{I}\} & : f = t. \end{cases}$$

That is, $D(e,f)$ is the set of independent sets which give rise to the arc ef. For each arc ef of D with $e \neq s$, fix a total ordering $\prec$ of $D(e,f)$. Notice that the crude augmentation is defined with respect to just one element $J \in D(e,f)$ for each arc ef of the augmenting path. Given an ordering on the elements of $D(e,f)$ for each arc ef, we assume that the crude augmentation chooses the first element when augmenting. For an independent set $J \in D(e,f)$, define the *level* $L_J(e,f)$ of J in ef to be

$$L_J(e,f) = \sum_{I \in D(e,f) : I \preceq J} \lambda_I.$$

Intuitively, we imagine each arc ef of Q as having $|D(e,f)|$ blocks stacked on top of one another. The bottom block corresponds to the first independent set J in the ordering of $D(e,f)$ and has height λ_J. The next block corresponds to the next independent set in the ordering, and so on. Thus, the level of J in ef is equal to the height of the block corresponding to J, plus the height of all the blocks corresponding to independent sets coming before J in the ordering of $D(e,f)$.

For each arc ef of Q, define the *capacity* of ef to be

$$u(e,f) = \begin{cases} x_f - y_f & : e = s \\ \sum_{J \in D(e,f)} \lambda_J & : e \neq s. \end{cases}$$

For an st-dipath $Q = s, e_1, e_2, .., e_k, t$, the *capacity* ε of Q is equal to the minimum of the capacities of its arcs. This is the height of a shortest stack of blocks along the path, or $x_e - y_e$, as the case may be. We seek an augmentation along Q of amount ε whose effect is to reduce the capacity of each arc of Q by ε, and to raise y_{e_1} by ε without affecting y_f for $f \neq e_1$.

In order to motivate the definition of the grand augmentation, consider the effect of a single crude augmentation using the chordless st-dipath Q. Due to the chordlessness of Q, for each arc ef of Q, the capacity of ef has decreased by exactly δ. If we consider some arc ef of Q, the first independent set $J \in D(e,f)$ has had its coefficient λ_J decreased by δ; intuitively, a piece of height δ has been cut from the bottom of the block corresponding to J in $D(e,f)$. For other arcs ab of Q, if $J \in D(a,b)$, then the block corresponding to J in $D(a,b)$ has been split into two blocks: one corresponding to J whose height has been reduced by δ, and (for the sake of illustration) directly below that, one corresponding to $J + e - f$ whose height is δ (we will not prove that $J + e - f \in D(a,b)$ here, and only mention this for the purposes of motivating the definition of the grand augmentation).

Performing another crude augmentation along Q will slice another piece from the bottom block of each stack, and so on. Let us call the independent sets involved in this process *mutations*. Specifically, if a set J' is the result of any number of alterations to a set J throughout the process, then J' is said to be a *mutation* of J.

Suppose that after some number of crude augmentations along Q, the capacity of each arc has been reduced by ε. With respect to the original coefficients λ, for an arc ef of Q, there are three types of sets in $D(e,f)$. First, there are sets J with $L_J(e,f) \leq \varepsilon$. For such a set, the block corresponding to J has been entirely sliced off the bottom of $D(e,f)$. Put another way, every mutation of J contains e but not f, for otherwise, some part of the block would remain. Secondly, there are sets $J \in D(e,f)$ with $\varepsilon < L_J(e,f) < \varepsilon + \lambda_J$. For these sets, the block corresponding to J has been reduced in height by $\lambda_J - L_J(e,f) + \varepsilon$. The remaining portion of the block may be split among some mutations of J, so some mutations of J contain e and some contain f. Last are the sets $J \in D(e,f)$ with $L_J(e,f) \geq \varepsilon + \lambda_J$. For such a set, the entire block for J remains, but again might be split among mutations of J.

Moreover, each mutation of such a J must contain f but not e.

We are now prepared to define the grand augmentation. Given a chordless st-dipath Q, let ε be the capacity of Q. For some $J \in \mathcal{I}$, define $C_J = \{(e,f) \in Q : J \in D(e,f)\}$, that is, C_J contains all arcs of Q which J gives rise to. For each independent set J such that $C_J \neq \emptyset$, let $\ell(J) = |C_J|$ and define an ordering $(e_1, f_1), (e_2, f_2), ..., (e_{\ell(J)}, f_{\ell(J)})$ on the elements of C_J such that $L_J(e_i, f_i) \leq L_J(e_{i+1}, f_{i+1})$ for all $1 \leq i \leq \ell(J) - 1$.

For each active independent set J, define $J^i = J\Delta\{e_1, f_1, e_2, f_2, ..., e_{i-1}, f_{i-1}\}$ for $i = 1$ to $\ell(J) + 1$, where, if $t = f_j$ for some j, we ignore it when taking the symmetric difference. Note that $J^1 = J$. We would like to have the property that

$$\sum_{i=1}^{\ell(J)+1} (\lambda'_{J^i} - \lambda_{J^i}) = 0, \tag{2.8}$$

that is, the value λ_J is distributed completely among the mutations of J. Consider again the three types of sets $J \in D(e_i, f_i)$. If $L_J(e_i, f_i) \leq \varepsilon$, then every mutation of J should contain e_i but not f_i. Thus, J^i should get a coefficient of 0 for such a J and i. If $\varepsilon < L_J(e_i, f_i) < \lambda_J + \varepsilon$, then the coefficient of J^i should be chosen as $L_J(e_i, f_i) - \varepsilon - \sum_{j<i} \lambda'_{J^j}$ so that the coefficients of the mutations of J containing f_i but not e_i sum to $L_J(e_i, f_i) - \varepsilon$. If $L_J(e_i, f_i) > \lambda_J + \varepsilon$, then every mutation of J should contain f_i but not e_i, so J^i should get the coefficient $\lambda_J - \sum_{j<i} \lambda'_{J^j}$. Formally, define the coefficient of J^1 as

$$\lambda'_{J^1} = \max\left(0, \min\left(\lambda_J, L_J(e_1, f_1) - \varepsilon\right)\right).$$

For $i > 1$, let the coefficient of J^i be given by

$$\lambda'_{J^i} = \lambda_{J^i} + \max\left(0, \min\left(\lambda_J, L_J(e_i, f_i) - \varepsilon\right) - \sum_{1 \leq j < i} \lambda'_{J^j} + \sum_{2 \leq j < i} \lambda_{J^j}\right). \tag{2.9}$$

Notice that $L_J(e_{\ell(J)+1}, f_{\ell(J)+1})$ is always undefined, so we will take the convention that $L_J(e_{\ell(J)+1}, f_{\ell(J)+1}) = \infty$. The above definition, along with this convention, guarantees that the property (2.8) is satisfied. We now show that the grand augmentation has properties similar to the crude augmentation. The second part of the theorem is particularly important, as it allows us to place a polynomial bound on the number of active independent sets.

Theorem 2.3.1. *Suppose λ' is obtained from λ by a grand augmentation along the st-dipath $Q = s, e_1, e_2, ..., e_m, t$ with capacity ε. Then λ' satisfies (2.2) and $y' = y + \varepsilon \{e_1\}$. Moreover, $|\{J \in \mathcal{I} : \lambda'_J > 0\}| \leq |\{J \in \mathcal{I} : \lambda_J > 0\}| + m$.*

Proof. We first show that λ' satisfies (2.2). This is the case if

$$\sum_{i=1}^{\ell(J)+1} \lambda'_{J^i} = \sum_{i=1}^{\ell(J)+1} \lambda_{J^i}.$$

By the convention that $L_J(e_{\ell(J)+1}, f_{\ell(J)+1}) = \infty$, it suffices to show that

$$\sum_{i=1}^{j} \lambda'_{J^i} \leq \sum_{i=1}^{j} \lambda_{J^i} \tag{2.10}$$

for $j = \ell(J)$. We prove that (2.10) holds for $j = 1, ..., \ell(J)$ by induction on j. For $j = 1$, the assertion is trivial. For $j > 1$,

$$\begin{aligned}\sum_{i=1}^{j} \lambda'_{J^i} &= \sum_{i=1}^{j-1} \lambda'_{J^i} + \lambda'_{J^j} \\ &= \sum_{i=1}^{j-1} \lambda'_{J^i} + \lambda_{J^j} + \max\left(0, \min\left(\lambda_J, L_J(e_j, f_j) - \varepsilon\right) - \sum_{i=1}^{j-1} \lambda'_{J^i} + \sum_{i=2}^{j-1} \lambda_{J^i}\right).\end{aligned}$$

If the maximum in the above expression is equal to 0, then the assertion follows from the inductive hypothesis. Otherwise,

$$\begin{aligned}\sum_{i=1}^{j} \lambda'_{J^i} &= \sum_{i=1}^{j-1} \lambda'_{J^i} + \lambda_{J^j} + \min\left(\lambda_J, L_J(e_j, f_j) - \varepsilon\right) - \sum_{i=1}^{j-1} \lambda'_{J^i} + \sum_{i=2}^{j-1} \lambda_{J^i} \\ &= \sum_{i=2}^{j} \lambda_{J^i} + \min\left(\lambda_J, L_J(e_j, f_j) - \varepsilon\right) \\ &\leq \sum_{i=1}^{j} \lambda_{J^i},\end{aligned}$$

as required.

We now show that $y' = y + \varepsilon \{e_1\}$. It is easy to see that $y'_u = y_u$ for vertices u not on Q. Consider $e = e_k$ for some $1 \leq k \leq m$. We first calculate

$$\text{up}(e, \lambda, \lambda') = \text{up}(e) = \sum_{J^i : e \in J^i \setminus J} (\lambda'_{J^i} - \lambda_{J^i}).$$

Intuitively, up(e) is the amount that y_e is increased by a grand augmentation; this happens whenever a mutation of an independent set J with $e \notin J$ contains e.

Notice that $e \in J^i \setminus J$ only if $J \in D(e, f)$, where ef is the arc of Q with e as its tail. Therefore, up(e) depends only on the mutations of the sets $J \in D(e, f)$. For such a set J, let

$$P_{J,e} = P_J = \sum_{J^i : e \in J^i} \left(\lambda'_{J^i} - \lambda_{J^i}\right).$$

Evidently, up$(e) = \sum(P_J : J \in D(e, f))$. Let $J \in D(e, f)$ and let (e_1, f_1), (e_2, f_2),..., $(e_{\ell(J)}, f_{\ell(J)})$ be the ordering of C_J. Then $(e, f) = (e_r, f_r)$ for some r, $1 \leq r \leq \ell(J)$. We consider three cases for J.

First, if $L_J(e_r, f_r) \leq \varepsilon$, then $L_J(e_s, f_s) \leq \varepsilon$ for all $1 \leq s \leq r$ by the ordering defined on C_J in the grand augmentation. Hence for all mutations of the form J^s for $1 \leq s \leq r$, the maximum in (2.9) is 0. Therefore, $\lambda'_{J^1} = 0$ and for $2 \leq s \leq r$, $\lambda'_{J^s} = \lambda_{J^s}$. It follows from (2.8) that

$$\lambda_J = \lambda_{J^1} = \sum_{s : r < s \leq \ell(J)+1} (\lambda'_{J^s} - \lambda_{J^s}),$$

and hence $P_J = \lambda_J$, since every mutation J^s in the sum above contains e but not f.

Next suppose $L_J(e_r, f_r) \geq \varepsilon + \lambda_J$. In this case, the minimum in (2.9) is λ_J. Therefore,

$$\lambda'_{J^r} = \lambda_{J^r} + \lambda_J - \sum_{s : 1 \leq s < r} \lambda'_{J^s} + \sum_{s : 2 \leq s < r} \lambda_{J^s},$$

and it follows that

$$0 = \sum_{s : 1 \leq s \leq r} (\lambda'_{J^s} - \lambda_{J^s}), \text{ which implies } 0 = \sum_{r < s \leq \ell(J)+1} (\lambda'_{J^s} - \lambda_{J^s}).$$

Since $\lambda'_{J^s} \geq \lambda_{J^s}$ $(s \neq 1)$, equality must hold for all $r < s \leq \ell(J)+1$. As all mutations in the first sum above contain f but not e, $P_J = 0$.

Finally, consider the case $\varepsilon < L_J(e_r, f_r) < \varepsilon + \lambda_J$. The maximum in (2.9) is

$$L_J(e_r, f_r) - \varepsilon - \sum_{s:1\leq s<r} \lambda'_{J^s} + \sum_{s:2\leq s<r} \lambda_{J^s}.$$

Note that this value is actually the minimum in (2.9), but it is always non-negative, and hence is also the maximum. This implies that

$$\sum_{s:1\leq s\leq r} (\lambda'_{J^s} - \lambda_{J^s}) = L_J(e_r, f_r) - \varepsilon - \lambda_J.$$

It follows from (2.8) that

$$P_J = \sum_{s:r<s\leq \ell(J)+1} (\lambda'_{J^s} - \lambda_{J^s}) = \lambda_J + \varepsilon - L_J(e_r, f_r).$$

Now, let I be that set in $D(e,f)$ with $L_I(e,f) \geq \varepsilon$ but $L_J(e,f) < \varepsilon$ for all $J \in D(e,f)$ with $J < I$. We have

$$\mathrm{up}(e) = \lambda_I + \varepsilon - L_I(e,f) + \sum_{J\in D(e,f), J<I} \lambda_J = \varepsilon$$

A similar argument shows that, for vertices $e \neq e_1$ of Q, $\mathrm{down}(e) = \varepsilon$, where

$$\mathrm{down}(e) = \sum_{J^i : e\in J\setminus J^i} (\lambda'_{J^i} - \lambda_{J^i}).$$

Since $\{J^i : e_1 \in J \setminus J^i\} = \emptyset$, $\mathrm{down}(e_1) = 0$. It follows that $y' = y + \varepsilon\,\{e_1\}$.

For the second part of the theorem, let $A = \bigcup_{ef\in Q} D(e,f)$. Note that $\lambda_J > 0$ for all $J \in A$. Consider

$$|\,\{J \in \mathcal{I} : \lambda'_J > 0\}\,| \leq |\,\{J \in \mathcal{I}\setminus A : \lambda_J > 0\}\,| + |\,\left\{J^i : J \in A, \lambda'_{J^i} > 0\right\}|.$$

Now,

$$\begin{aligned}
&|\,\{J^i : J \in A, \lambda'_{J^i} > 0\}\,| \\
=&\sum_{J\in A} |\,\{J^i : \lambda'_{J^i} > 0\}\,| \\
\leq&\sum_{J\in A} (1 + |\,\{ef \in Q : ef \in C_J,\ \varepsilon < L_J(e,f) < \varepsilon + \lambda_J\}\,|) \\
=&|A| + \left|\bigcup_{J\in A} \{ef \in Q : ef \in C_J,\ \varepsilon < L_J(e,f) < \varepsilon + \lambda_J\}\right| \\
\leq&|A| + m.
\end{aligned}$$

The first inequality above holds since each $J \in A$ is replaced by one set, plus at most one extra set for every edge $ef \in C_J$ with $\varepsilon < L_J(e,f) < \varepsilon + \lambda_J$. The second equality holds since each arc ef of Q can satisfy $\varepsilon < L_J(e,f) < \varepsilon + \lambda_J$ for at most one set $J \in A$, and the last inequality holds since there are at most m arcs which can satisfy the same. It follows that

$$|\,\{J \in \mathcal{I} : \lambda'_J > 0\}\,| \leq |\,\{J \in \mathcal{I} : \lambda_J > 0\}\,| + m$$

as required. □

Theorem 2.3.1 verifies that our convex combination is valid, providing that the mutations we have defined are independent sets. To prove this, we require two results, which are the subject of the next section. Before moving on, we determine the running time of the grand augmentation.

Lemma 2.3.2. *Given λ, the auxiliary digraph can be constructed, a chordless st-dipath Q can be selected, and a grand augmentation can be performed along Q in time $O(n^2 \cdot q \cdot p)$, where $O(q)$ is the complexity of the independence oracle, and $O(p)$ is the number of active independent sets.*

Proof. The time to build the auxiliary digraph clearly dominates. This involves testing, for each potential arc ef of the auxiliary digraph, whether there exists some active J such that $f \in C(J,e)$. Furthermore, note that the grand augmentation actually needs to determine the sets $D(e,f)$ for each arc ef of D. So indeed, for each potential arc, we will have to test whether it is an arc with respect to each active J. For each such J, this involves $O(n^2)$ tests of independence. Hence, where $O(q)$ is the running time of the independence oracle and the number of active independent sets, the time to build the auxiliary digraph is $O(n^2 \cdot q \cdot p)$. □

2.4 Two Lemmas

We wish to show that the grand augmentation is indeed valid. This only requires us to show that each mutation is an independent set. The results in this section will establish just that. However, looking ahead, we will also have to place a bound on the running time of the algorithm. Lemma 2.3.2 is an important part of that bound, but we still have the problem of determining what exactly $O(p)$ is, that is, we must bound the number of positive components of λ. From Theorem 2.3.1, we need only prove that the number of grand augmentations required to solve an instance is polynomially bounded. Since each augmentation adds a polynomial number of (newly) positive components to λ, we obtain a polynomial bound.

Central to the proof of a polynomial number of grand augmentations are necessary preconditions for the appearance of a new arc in the auxiliary digraph. In addition to proving that the mutations are independent, the main business of the lemmas in this section is to establish those necessary preconditions.

Note that for each mutation J' of J, either $|J| = |J'|$ or $|J| + 1 = |J'|$. These correspond to, respectively, mutations involving only arcs ef, where $e, f \in S$, and mutations involving an arc et. The first two results apply to the first type of mutation.

Proposition 2.4.1. *Let $(S, \mathcal{I})$ be a matroid and let $J \in \mathcal{I}$. Let a_1,b_1,a_2,b_2,..., a_p,b_p be a sequence of distinct elements of S with $a_1, ..., a_p \notin J$ and $b_1, ..., b_p \in J$ such that*

(i) $b_i \in \mathrm{C}(J, a_i)$ for $1 \leq i \leq p$ and

(ii) $b_j \notin \mathrm{C}(J, a_i)$ for $1 \leq i < j \leq p$.

Let $J' = J \Delta \{a_1, b_1, ..., a_p, b_p\}$. Then $J' \in \mathcal{I}$ and $\mathrm{cl}(J') = \mathrm{cl}(J)$.

Proof. Let $X = J \cup \{a_1, ..., a_p\}$. By (i), $\mathrm{C}(J, a_i)$ exists for all $1 \leq i \leq p$, and hence we have $a_i \in \mathrm{cl}(J)$. Therefore, $\{a_1, ..., a_p\} \subseteq \mathrm{cl}(J)$, and so $\mathrm{cl}(J) = \mathrm{cl}(X)$ by Lemma 1.3.8 and $r(J) = r(X)$ by Lemma 1.3.6. By condition (ii), $\mathrm{C}(J, a_i) \subseteq J \cup \{a_i\} \setminus \{b_p, ..., b_{i+1}\} \subseteq J' \cup \{b_1, ..., b_i\}$. Since $C(J, a_i) \subseteq J' \cup \{b_1, ..., b_i\}$ and $b_i \in C(J, a_i)$, it follows that $C(J, a_i) \nsubseteq J' \cup \{b_1, ..., b_{i-1}\}$. Consequently, $b_i \in \mathrm{cl}(J' \cup \{b_1, ..., b_{i-1}\})$. It follows by Lemma 1.3.8 that $\mathrm{cl}(J' \cup \{b_1, ..., b_i\}) = \mathrm{cl}(J' \cup \{b_1, ..., b_{i-1}\})$ for all $1 \leq i \leq p$. Therefore $\mathrm{cl}(J') = \mathrm{cl}(X)$ and by Lemma 1.3.6, $r(J') = r(X)$. We conclude that $\mathrm{cl}(J) = \mathrm{cl}(J')$ and $|J| = r(J) = r(J') = |J'|$, so $J' \in \mathcal{I}$, as required. □

This second lemma gives us the preconditions for the appearance of a new arc in the auxiliary digraph. This will be important in proving the bound on the number of grand augmentations.

Lemma 2.4.2. *Let $(S, \mathcal{I})$ be a matroid and let $J \in \mathcal{I}$. Let $a_1, b_1, ..., a_p, b_p$ be a sequence of distinct elements of S with $a_1, ..., a_p \notin J$ and $b_1, ..., b_p \in J$ such that*

(i) $b_i \in \mathrm{C}(J, a_i)$ *for* $1 \leq i \leq p$ *and*

(ii) $b_j \notin \mathrm{C}(J, a_i)$ *for* $1 \leq i < j \leq p$.

Let $J' = J \Delta \{a_1, b_1, ..., a_p, b_p\}$. If $f \in \mathrm{C}(J', e)$ and ($e \in J$ or $J + e - f \notin \mathcal{I}$), then there exist k and ℓ with $1 \leq k \leq \ell \leq p$ such that $f \in \mathrm{C}(J, a_k)$ and ($b_\ell = e$ or $b_\ell \in \mathrm{C}(J, e)$).

Proof. Let $J_t = J' \Delta \{a_1, b_1, ..., a_t, b_t\}$ for $0 \leq t \leq p$. If $e \in J$, then since $\mathrm{C}(J', e)$ exists, $e \notin J'$ and so $e = b_\ell$ for some $1 \leq \ell \leq p$. Otherwise, choose the least ℓ such that $\mathrm{C}(J, e) = \mathrm{C}(J_i, e)$ for all $\ell \leq i \leq p$. Then $\mathrm{C}(J_{\ell-1}, e) \neq \mathrm{C}(J_\ell, e) = \mathrm{C}(J_{\ell-1} - a_\ell + b_\ell, e)$, so $b_\ell \in \mathrm{C}(J_\ell, e)$ by Lemma 1.3.3. Moreover, $\mathrm{C}(J_\ell, e) = \mathrm{C}(J, e)$, so $b_\ell \in \mathrm{C}(J, e)$.

Now, choose the greatest k such that $f \in \mathrm{C}(J_i, e)$ for all $0 \leq i < k$. Then $f \in \mathrm{C}(J_{k-1}, e) \setminus \mathrm{C}(J_k, e)$. As $e \in \mathrm{C}(J_{k-1}, e) \cap \mathrm{C}(J_k, e)$, by the Strong Circuit Axiom (Lemma 1.3.4) there exists a circuit C such that

$$\begin{aligned} f \in C &\subseteq (\mathrm{C}(J_{k-1}, e) \cup \mathrm{C}(J_k, e)) \setminus \{e\} \\ &\subseteq J_{k-1} + b_k. \end{aligned}$$

Thus, $C = \mathrm{C}(J_{k-1}, b_k)$. By condition (ii),

$$\begin{aligned} \mathrm{C}(J, a_k) &\subseteq J \cup \{a_k\} \setminus \{b_{k+1}, ..., b_p\} \\ &\subseteq J_{k-1} + b_k. \end{aligned}$$

Hence, $f \in C = \mathrm{C}(J_{k-1}, b_k) = \mathrm{C}(J, a_k)$. If $e \in J$, then $\mathrm{C}(J_i, e)$ does not exist for $\ell \leq i \leq p$, so we must have $k \leq \ell$. Otherwise, since $f \notin \mathrm{C}(J_i, e)$ for $\ell \leq i \leq p$, we also have $k \leq \ell$. $\square$

The following lemma deals with the case where the mutation has one more element than the original independent set. As above, this lemma shows that the mutation is an independent set, and gives identical preconditions for a new arc to appear after the augmentation.

Lemma 2.4.3. *Let $(S, \mathcal{I})$ be a matroid and let $J \in \mathcal{I}$. Let $a_1, b_1, ..., a_p, b_p,\ a_{p+1}$ be a sequence of distinct elements of S with $a_1, ..., a_p, a_{p+1} \notin J$ and $b_1, ..., b_p \in J$ such that*

(i) $b_i \in \mathrm{C}(J, a_i)$ *for* $1 \leq i \leq p$,

(ii) $b_j \notin \mathrm{C}(J, a_i)$ *for* $1 \leq i < j \leq p$ *and*

(iii) $J + a_{p+1} \in \mathcal{I}$.

Let $J' = J \Delta \{a_1, b_1, ..., a_p, b_p, a_{p+1}\}$. Then $J' \in \mathcal{I}$. Moreover, if $f \in \mathrm{C}(J', e)$ and ($e \in J$ or $J + e - f \notin \mathcal{I}$), then there exist k and ℓ with $1 \leq k \leq \ell \leq p$ such that $f \in \mathrm{C}(J, a_k)$ and ($b_\ell = e$ or $b_\ell \in \mathrm{C}(J, e)$).

Proof. By Proposition 2.4.1, we have $\mathrm{cl}(J' - a_{p+1}) = \mathrm{cl}(J)$, and hence $J' \in \mathcal{I}$. It is easy to see that $J + a_{p+1}$ and the sequence $a_1, b_1, ..., a_p, b_p$ satisfy (i) and (ii) of Lemma 2.4.2. If $e \in J$, then $e \in J + a_{p+1}$. If $J + e - f \notin \mathcal{I}$, then $J + a_{p+1} + e - f \notin \mathcal{I}$. Hence, the hypothesis of Lemma 2.4.2 is satisfied with respect to $J + a_{p+1}$ and the sequence $a_1, b_1, ..., a_p, b_p$. From Lemma 2.4.2, we have k and ℓ with $1 \leq k \leq \ell \leq p$ such that $f \in \mathrm{C}(J + a_{p+1}, a_k) = \mathrm{C}(J, a_k)$ and $b_\ell = e$ or $b_\ell \in \mathrm{C}(J + a_{p+1}, e)$. If $b_\ell \in \mathrm{C}(J + a_{p+1}, e)$, then $e \notin J$, so $C(J, e) = C(J + a_{p+1}, e)$. □

There are lemmas very similar to those seen in this section found in Cunningham's paper [3] on this topic, where they appear as Lemma 4.3 and Lemma 4.4. For Lemma 4.3 (which corresponds to Lemma 2.4.2), the antecedent of the conclusion to the Lemma reads as follows:

$$\text{If } f \in C(J', e) \text{ and } f \notin C(J, e).$$

Cunningham also notes that, if $f \in C(J', e)$ and $C(J, e)$ does not exist, then we consider the antecedent to be satisfied. Thus, we can restate Cunningham's antecedent as

$$\text{If } f \in C(J', e) \text{ and } (J + e - f \notin \mathcal{I} \text{ or } J + e \in \mathcal{I} \text{ or } e \in J).$$

This is in contrast to our stronger antecedent, which reads

$$\text{If } f \in C(J', e) \text{ and } (J + e - f \notin \mathcal{I} \text{ or } e \in J).$$

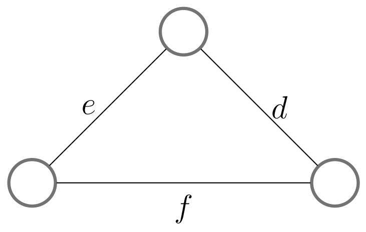

Figure 2.1: Counterexample to Lemma 4.4 of [3]

This change is not without reason; indeed, it is possible to construct an example which satisfies $f \in C(J', e)$ and $J + e \in \mathcal{I}$ without satisfying the consequent of the conclusion. Consider the following example.

Let M be the graphic matroid of the graph in Figure 2.1, let $J = \{f\}$, and let $J' = \{d, f\}$. In Lemma 2.4.3, this corresponds to a sequence of length one with $a_1 = d$. Then $f \in C(J', e)$ and $J + e \in \mathcal{I}$ while $e \notin J$ and $J + e - f \in \mathcal{I}$, so this example does not satisfy Lemma 2.4.3 but it does satisfy the corresponding version in Cunningham's paper. Moreover, since $f \notin C(J, e)$, the conclusion of Lemma 2.4.3 is not satisfied. In Cunningham's paper [3] the consequent of the implication reads

there exists k, ℓ with $1 \leq k \leq \ell \leq p$ such that $f \in C(J, a_k)$ and $b_\ell = e$ or $b_\ell \in C(J, e)$ or $e \notin J$ and $J + e \in \mathcal{I}$.

In the example above, this difference is not particularly relevant. Since $f \notin C(J, a_k)$, the conclusion cannot be satisfied in this case. It follows that there must be an error in Cunningham's proof of Lemma 4.4 (which corresponds to Lemma 2.4.3 here).

Note that all counterexamples will have to satisfy $f \in C(J', e)$ and $J + e \in \mathcal{I}$ and $e \notin J$. This necessarily implies that $C(J', e)$ exists, that is, $e \in \mathrm{cl}(J')$ but $e \notin \mathrm{cl}(J)$. By Proposition 2.4.1, it must be the case that $|J'| = |J| + 1$ (since the closure does not change if the sets are equicardinal). It follows that all counterexamples occur in the case where our sequence has the form a_1, b_1, ...,a_p, b_p, a_{p+1}, or in other words, that Lemma 4.3 of [3] is correct and is equivalent to Lemma 2.4.2.

In a subsequent paper [5], Cunningham again makes use of Lemma 4.3 and Lemma 4.4 from [3], in order to give a faster algorithm for the matroid intersection problem. This is the subject of the next chapter. The error in these results which we have just examined was first pointed out by Haselmayr who, as part of his diploma thesis [14], fixed this problem and gave a complete proof of correctness of the improved matroid intersection algorithm. His fix is the same as the fix presented in this section, though his proof of Lemma 2.4.2 is lengthy and relies on induction, as Cunningham's proof does. The simpler constructive proof presented in this section is, to our knowledge, a new proof of this result.

To see that the mutations are independent sets, we can use Lemma 2.4.2 or Lemma 2.4.3. For some mutation J^i of J, we simply take the arcs (a_1, b_1), (a_2, b_2), ..., (a_{i-1}, b_{i-1}) of C_J in the order that they appear on Q. If some $b_j = s$ for $j < i$, then we use Lemma 2.4.3; otherwise, we use Lemma 2.4.2. Due to the chordlessness of Q, these arcs, in the specified order, satisfy the appropriate lemma with respect to the independent set J. Thus, the mutations are all independent sets, so the grand augmentation produces a convex combination of independent sets with the properties given in Theorem 2.3.1.

2.5 Analysis

So far, we have defined the grand augmentation and proven it correct. We require one more definition to give a statement of the algorithm. For this section, and in the algorithm, we define an ordering on the vertices of D as $S = \{1, 2, 3, ..., n\}$ with $s = 0$ and $t = n + 1$. Consider Algorithm 2, which requires a matroid $M = (S, \mathcal{I})$ and a vector $x \in \mathbb{R}^S$ as input.

Algorithm 2 Matroid Polyhedron Membership Algorithm

```
$\lambda \leftarrow 0$
$\lambda_\emptyset \leftarrow 1$
$D \leftarrow D(M, x, \lambda)$
**while** there exists an $st$-dipath in $D$ **do**
    $Q \leftarrow$lexicographically least shortest $st$-dipath in $D$
    $\lambda \leftarrow$result of grand augmentation on $Q$
    $D \leftarrow D(M, x, \lambda)$
**end while**
$A \leftarrow \{e \in S :$ there exists an $se-$dipath in $D\}$
**return** $(\lambda, A)$
```

We still have said little about the running time of the algorithm. In order to understand the approach, it is helpful to consider an analogy to network flows. In the Ford-Fulkerson algorithm [11], given a flow x, the algorithm finds an x-augmenting path in the residual digraph, and performs an augmentation of amount equal to the capacity of the path. As Ford and Fulkerson observed, there exist flow networks for which the algorithm does not terminate. The grand augmentation is roughly equivalent to the augmentation step in the Ford-Fulkerson algorithm. The analogy is not perfect, however, as we are not aware of any instances in which iteratively choosing any chordless st-dipath and performing a grand augmentation leads to infinite computation.

Where the analogy is fruitful is in considering the solutions to this problem with the Ford-Fulkerson algorithm. Dinic [6] and, independently, Edmonds and Karp [9] proposed a simple, elegant solution. By iteratively selecting a *shortest* x-augmenting path, the Edmonds-Karp algorithm achieves a running time of $O(|V||E|^2)$ for a flow network (V, E). The analogy is again imperfect, since our analog is to choose a lexicographically least shortest st-dipath; the idea of choosing such a path is originally due to Schönsleben [20], and independently, Lawler and Martel [18]. Nonetheless, the Edmonds-Karp algorithm, and in particular, its analysis, guides our analysis of Algorithm 2.

Let (V, E) be a flow network and $c \in \mathbb{R}^E$ be arc capacities. Let $s, t \in V$ be a source and a sink, respectively. For a flow x, let $d_x(u, v)$ denote the length of a shortest x-incrementing uv-path in (V, E) (∞ if none exists). Let x ($0 \leq x \leq c$) be an st-flow in (V, E). The analysis in [9] proceeds as follows. First, if an arc uv appears in the residual digraph after augmenting x to x', then vu was part of the x-augmenting path chosen. Using this, it is possible to show that the lengths of each path chosen by the Edmonds-Karp algorithm, in the order in which they are chosen, form a nondecreasing sequence. We call this a *monotonicity property*. Notice that, by the monotonicity property and since an arc ef appears in the residual digraph only if fe was part of the augmenting path, any arc appearing after an augmentation on a path of length k cannot itself be part of an augmenting path of length k. Next, we observe that an augmentation always destroys some arc of the residual digraph. Combining this with the previous observation, we can deduce that there are $O(|E|)$ augmentations using paths of a fixed length. Such a path can be found in $O(|E|)$ time by breadth-first search, and there are $O(|V|)$ distinct lengths of augmenting paths, giving the bound.

Due to the manner in which arcs can appear in the auxiliary digraph, the analysis of the Edmonds-Karp algorithm takes some work to generalize to the present context. However, it still serves as something of a template for how we will proceed. We start by proving that, after a grand augmentation along an st-dipath Q, some arc of Q is not an arc of the next auxiliary digraph. Next we prove a technical lemma which serves to translate the necessary preconditions for the appearance of an arc in the auxiliary digraph into a more useful form. This facilitates a proof of our analog to the monotonicity property. Following this, we make a small digression to present a counterexample to a Lemma in [3]. We go on to prove several results about the successive auxiliary digraphs examined by Algorithm 2 when the length of a shortest st-dipath in those digraphs is constant. Using these results, we give a polynomial bound on the running time of Algorithm 2. The argument in this section closely resembles that presented by Cunningham [3], though it has been simplified.

For the remainder of the section, let $M = (S, \mathcal{I})$ and $x \in \mathbb{R}^S$ be an instance of the matroid polyhedron membership problem. Run Algorithm 2 with M and x as inputs. Let D_0, D_1,..., D_k be the sequence of auxiliary digraphs constructed by Algorithm 2, and let $d_0, d_1, ..., d_k$ be their distance functions, respectively. Let Q_i be the lexicographically least shortest st-dipath in D_i for $0 \leq i < k$. Let λ_0 be the initial λ before entering the main loop of Algorithm 2 and for $1 \leq i \leq k$, let λ_i be the result of a grand augmentation along the path Q_{i-1} in the auxiliary digraph $D_{i-1} = D(M, x, \lambda_{i-1})$.

We call an arc ab of an st-dipath Q *critical* if the capacity $u(a, b)$ of ab is equal to the capacity of Q.

Lemma 2.5.1. *For all i, $0 \leq i < k$, if ef is a critical arc of Q_i, then ef is not an arc of D_{i+1}.*

Proof. Suppose ef is a critical arc of Q_i which is also an arc of D_{i+1}. Let $y = y(\lambda_i)$ and $y' = y(\lambda_{i+1})$. If $e = s$ then by Theorem 2.3.1, $y' = y + \varepsilon\{f\}$. In particular, $y'_f = x_f$ by the definition of $u(e, f)$, so we may assume $e \neq s$.

Since ef is a critical arc, for each $J \in D(e, f)$, every mutation of J contains e but not f. Hence, no such set can give rise to the arc ef. Then it must be that, for some arc $ab \neq ef$ of Q_i and for some $J \in D(a, b) \setminus D(e, f)$, there is a mutation J' of J for which $f \in C(J', e)$ while $f \notin C(J, e)$. It could be that $J + e \in \mathcal{I}$, in which case et is an arc of D, and since Q is chordless, $ef = et$. Then $J \in D(e, f)$, a contradiction. Hence $e \in J$ or $J + e - f \notin \mathcal{I}$ and $f \in C(J', e)$, so we can apply Lemma 2.4.2 or Lemma 2.4.3 to J, e, f, and the subsequence of arcs of Q which transform J into J'.

Since $J \notin D(e, f)$, ef cannot be one of the arcs of this subsequence. It follows that the lemma gives us some p and q with p preceding q on Q such that $f \in C(J, p)$ and $q \in C(J, e)$. Since Q is chordless and pf is an arc of D, it must be that f (and hence also e) precedes p on Q. Similarly, as eq is an arc of D, it must be that q precedes e (and hence also f) on Q. But now, e precedes p, p precedes q, and q precedes e on Q, a contradiction. □

For some i, $0 \leq i < k$, let $D = D_i$, $\lambda = \lambda_i$, $Q = Q_i$, and similarly, $D' = D_{i+1}$, and $\lambda' = \lambda_{i+1}$. Let $d = d_i$ and $d' = d_{i+1}$. In Cunningham's paper [3], the monotonicity property (Lemma 3.2) is stated as

For all $f \in S$, $d'(s, f) \geq d(s, f)$ and $d'(f, t) \geq d(f, t)$.

This relies on the following lemma (which is no longer true, in light of the previous section) which appears as Lemma 4.6 in [3].

If ef is an arc of D' but not of D, then $e, f \in S$ and there exist vertices a, b of Q with a preceding b on Q such that ($a = f$ or af is an arc of D) and ($b = e$ or eb is an arc of D).

Since the proof of this lemma follows from Cunningham's version of Lemma 2.4.2 or Lemma 2.4.3, it is true whenever those results hold. The case where they do not hold is, as we have seen, the case where $f \in C(J', e)$ and $J + e \in \mathcal{I}$. In this case, it is not hard to see that et is an arc of D. Hence, we have the following lemma.

Lemma 2.5.2. *If ef is an arc of D' but not of D, then $e, f \in S$ and one of the following holds.*

(a) There exist vertices a, b of Q with a preceding b on Q such that ($a = f$ or af is an arc of D) and ($b = e$ or eb is an arc of D).

(b) et is an arc of D.

Proof. Suppose ef is an arc of D' but not D. Since $y' \geq y$, $y'_e < x_e$ implies $y_e < x_e$, so we cannot have $e = s$. Similarly, for every mutation J' of J, either $\mathrm{cl}(J') = \mathrm{cl}(J)$ or $\mathrm{cl}(J') \supseteq \mathrm{cl}(J)$. In either case, $J' + a \in \mathcal{I}$ implies $J + a \in \mathcal{I}$. Thus, $e, f \in S$.

Since ef is an arc of D' but not D, there is some mutation J' of some set J with $\lambda_J > 0$ and $\lambda'_{J'} > 0$ such that $f \in C(J', e)$ while $f \notin C(J, e)$. If $J + e \in \mathcal{I}$, then et is an arc of

D, so assume not. Thus, either $e \in J$ or $J + e - f \notin \mathcal{I}$ and $f \in C(J', e)$. Therefore, we can apply Lemma 2.4.2 or Lemma 2.4.3 to the subsequence of arcs of Q which transform J into J'. From this, we obtain $a = a_k$ and $b = b_\ell$ which satisfy (a). □

We are now prepared to prove the monotonicity property.

Lemma 2.5.3. *For all $f \in S$,*

(i) if $d(s,f) < d(s,t)$, then $d'(s,f) \geq d(s,f)$, and

(ii) if $d(f,t) < d(s,t)$, then $d'(f,t) \geq d(f,t)$.

Proof. Suppose $d(s,f) < d(s,t)$ but $d'(s,f) < d(s,f)$, and choose f so that $d'(s,f)$ is minimized. Since $y' \geq y$, it must be that $d'(s,f) \geq 2$. Let e be the predecessor of f on some shortest sf-dipath in D'. Note that $d'(s,e) \geq d(s,e)$ by the choice of f. Now, since

$$d(s,f) > d'(s,f) = d'(s,e) + 1 \geq d(s,e) + 1,$$

it follows that ef is not an arc of D. Moreover, et is not an arc of D, since $d(s,f) < d(s,t)$. Therefore, there are vertices a, b of Q satisfying (a) of Lemma 2.5.2. In particular, a precedes b on Q, and Q is a shortest (chordless) path, so $d(s,a) < d(s,b)$. But now,

$$d(s,a) \geq d(s,f) - 1 \geq d'(s,f) = d'(s,e) - 1 \geq d(s,e) - 1 \geq d(s,b),$$

a contradiction. So (i) is proved, and the proof of (ii) is similar. □

There are, in fact, examples which demonstrate that the monotonicity lemma found in [3] is not always correct. Consider the graphic matroid $M = (S, \mathcal{I})$ of the graph in Figure 2.2.

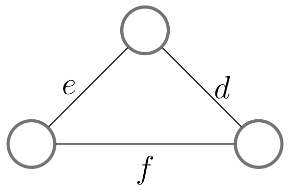

Figure 2.2: Counterexample to monotonicity lemma in [3].

Suppose we are given the point $x = (x_d, x_e, x_f) = (\frac{1}{2}, \frac{1}{2}, 1)$ and currently have a convex expression $y = \chi_J$ where $J = \{f\}$. The auxiliary digraph appears below.

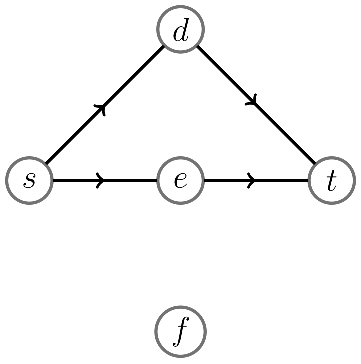

Figure 2.3: Auxiliary digraph with respect to M, x, y, and λ.

Note that $d(s, f) = \infty$. Choosing the path s, d, t, an augmentation will create a set $J' = \{d, f\}$ with coefficient $\frac{1}{2}$ and $J = \{f\}$ will have coefficient $\frac{1}{2}$ also. The auxiliary digraph with respect to this combination appears below.

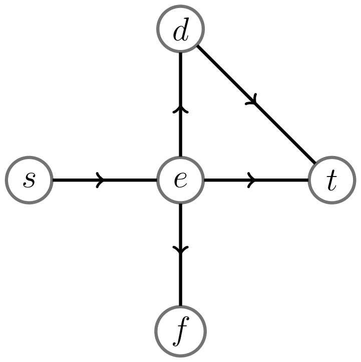

Figure 2.4: Auxiliary digraph after augmentation - note that $d'(s, f) < d(s, f)$.

Now, $d'(s, f) = 2$, contradicting the monotonicity lemma found in [3]. Note that since $d(s, f) > d(s, t)$, this example does not disagree with Lemma 2.5.3.

It follows from Lemma 2.5.3 that $d_j(s, t) \geq d_i(s, t)$ for all i and j with $0 \leq i < j \leq k$, that is, the sequence $d_0(s, t)$, $d_1(s, t)$, ..., $d_k(s, t)$ is monotone nondecreasing. Observe that any chordless st-dipath has at most $n + 2$ vertices, and hence at most $n + 1$ arcs; similarly, any chordless st-dipath has at least 3 vertices, and hence at least 2 arcs. Thus, there are at most n distinct values for $d_i(s, t)$ over all i, $0 \leq i < k$. Therefore, we can partition the sequence D_0,D_1,...,D_{k-1} into at most n contiguous subsequences, where, for all sets D_i, D_j within a subsequence, $d_i(s,t) = d_j(s,t)$.

With respect to Algorithm 2, the partition of D_0, D_1, ..., D_{k-1} above induces a partition of the augmentations performed, namely, for i between 2 and $n+1$, part i consists of all augmentations performed on augmenting paths of length i. For a fixed i, note that the augmentations in part i are necessarily consecutive augmentations of the algorithm. For each i, we refer to this set of augmentations (that is, part i) as *stage* i of Algorithm 2. This simplifies our analysis considerably; since we know that there are $O(n)$ stages, we can bound the total number of grand augmentations by bounding the number of grand augmentations within a stage. This is the purpose of the next three results.

Again, let $D = D_i$, $\lambda = \lambda_i$, $Q = Q_i$, $D' = D_{i+1}$, and $\lambda' = \lambda_{i+1}$ for some i, $0 \le i < k$. Let $d = d_i$ and $d' = d_{i+1}$. For each i, $0 \le i < k$, let H_i denote the subdigraph of D_i consisting of all vertices and arcs of D_i which belong to some shortest st-dipath in D_i. Let $H = H_i$ and $H' = H_{i+1}$. The following lemma gives necessary preconditions for an arc to appear on a shortest st-dipath inside a stage.

Lemma 2.5.4. *If $d(s,t) = d'(s,t)$ and ef is an arc of H' but not of H, then there exists an arc ab of Q such that af and eb are arcs of H.*

Proof. Suppose $d(s,t) = d'(s,t)$ and ef is an arc of H' but not H. By Lemma 2.5.3,

$$d(s,e) + d(f,t) \le d'(s,e) + d'(f,t) = d'(s,t) - 1.$$

In particular, equality must hold and so ef cannot be an arc of D or else it would also be an arc of H. Thus, we can apply Lemma 2.5.2. If et is an arc of D, then

$$d'(s,t) = d(s,t) = d(s,e) + 1 \le d'(s,e) + 1 = d'(s,f),$$

a contradiction. Hence, there must be vertices a, b of Q as in (a) of Lemma 2.5.2. Observe that $d(s,f) \ge d(s,e) + 1$ since $d(s,e) + d(f,t) = d(s,t) - 1$. Furthermore, we have $d(s,f) \le d(s,e) + 1$ since

$$d(s,f) \le d'(s,f) = d'(s,e) + 1 = d(s,e) + 1.$$

A similar argument shows that $d(e,t) = d(f,t) + 1$. Since a precedes b on Q, $d(s,a) + d(b,t) \le d(s,t) - 1$. Furthermore,

$$d(s,a) + d(b,t) \ge d(s,f) - 1 + d(e,t) - 1 = d(s,e) + d(f,t) = d(s,t) - 1.$$

It follows that $d(s,a) + d(b,t) = d(s,t) - 1$. In particular, this implies that $d(s,a) = d(s,f) - 1$ and $d(b,t) = d(e,t) - 1$, and so ab, af and eb are arcs of H. □

An observation that we will use in the next lemma is that, since af and ab are both arcs of H (as above), and Q is a lexicographically least shortest st-dipath which contains ab, it must be the case that $b < f$ in our ordering of N.

For each i, $0 \leq i < k$ and for each $e \in S + s$, define the *successor* $\sigma_i(e)$ of e to be the least f such that ef is an arc of H_i ($\sigma_i(e) = \infty$ if none exists). Note that, for each arc ef of the lexicographically least shortest st-dipath in H_i, $\sigma_i(e) = f$. For some i, $0 \leq i < k$, let $H = H_i$, $H' = H_{i+1}$, $\sigma = \sigma_i$, and $\sigma' = \sigma_{i+1}$. We first show that, inside a stage, and for each $e \in S$, the successor function is a nondecreasing one.

Lemma 2.5.5. *If $d(s,t) = d'(s,t)$, then for each $e \in S + s$, $\sigma'(e) \geq \sigma(e)$.*

Proof. Suppose not. Then for some f, $f = \sigma'(e) < \sigma(e)$. Since $\sigma(e) \neq f$, it must be that ef is an arc of H' but not H. By Lemma 2.5.4, there is an arc ab of Q such that af and eb are arcs of H. Now,

$$f < \sigma(e) \leq b = \sigma(a) \leq f,$$

a contradiction. □

We can now bound the number of grand augmentations performed in a stage.

Lemma 2.5.6. *For a fixed ℓ, $2 \leq \ell \leq n+1$, Algorithm 2 performs $O(n^2)$ grand augmentations using paths of length ℓ. Consequently, there are $O(n^3)$ digraphs in the sequence D_0, D_1, ..., D_k.*

Proof. As we have already observed, the number of distinct lengths of st-dipaths that can occur in the sequence is n, so the second part of the lemma follows immediately from the first.

To prove the first part, consider a stage having dipaths of length ℓ for some $2 \leq \ell \leq n+1$. Let D_p, ..., D_q, $p \leq q$, be the subsequence of D_0, ..., D_k corresponding to this stage. Consider the sequence of vectors $(\sigma_i(e) : e \in S + s)$ for $i = p$ to q. By Lemma 2.5.5, the sequence is componentwise nondecreasing. Moreover, since each component has a value between 1 and $n+1$, there are $O(n^2)$ distinct vectors in the sequence. By Lemma

2.5.1 and Lemma 2.5.5, for each i, $p \leq i < q$, there exists some $e \in S + s$ such that $\sigma_i(e) < \sigma_{i+1}(e)$. It follows that each vector in the sequence is distinct, and hence there are $O(n^2)$ augmentations using paths of length ℓ. □

At last, we can put the pieces together to prove that Algorithm 2 is polynomial time.

Theorem 2.5.7. *Given a matroid $M = (S, \mathcal{I})$ and a vector $x \in \mathbb{R}^S$, Algorithm 2 returns (λ, A) such that (y, A) is an optimal pair in the Membership Min-Max Theorem (2.1.2). Moreover, Algorithm 2 is a polynomial time matroid algorithm with running time $O(n^9 \cdot q)$.*

Proof. By Theorem 2.3.1 and the Augmenting Path Theorem (2.2.1), the certificate (λ, A) returned by Algorithm 2 is valid. By Lemma 2.5.6, there are $O(n^3)$ grand augmentations. By Lemma 2.3.2, each such augmentation is computed in $O(n^2 \cdot pq)$ operations, where $O(p)$ is the number of active independent sets. By Theorem 2.3.1, each grand augmentation increases the number of active sets by $O(n)$. Since there are $O(n^3)$ augmentations, $p \in O(n^4)$. It follows that Algorithm 2 terminates in $O(n^9 \cdot q)$ operations. □

The running time of the algorithm can easily be improved by reducing the number of active independent sets to at most $n + 1$ after each grand augmentation, per Carathéodory's Theorem. Such a reduction can be performed in $O(n^3)$ operations, and with this modification, there are $O(n)$ active independent sets. By Lemma 2.3.2, each grand augmentation would then require $O(n^3 \cdot q)$ operations, for a total running time of $O(n^6 \cdot q)$. However, this modification sacrifices the following property of λ.

Theorem 2.5.8. *The vector $\lambda \in \mathbb{R}^{\mathcal{I}}$ returned by Algorithm 2 satisfies the following properties.*

(i) every nonzero component of λ is an integer combination of the values $\{x_e : e \in S\} \cup \{1\}$.

(ii) there are $O(n^4)$ nonzero components of λ.

Proof. Property (ii) holds by Theorem 2.3.1. We claim that the following property is invariant under the action of the algorithm. For each active J, the coefficient λ_J is an integer combination of the values $\{x_e : e \in S\} \cup \{1\}$.

Evidently, the property holds at initialization. Consider a grand augmentation along the st-dipath $Q = s, e_1, e_2, ..., e_k, t$, and suppose the property holds before the augmentation.

Recall that, for an arc e_ie_{i+1} of Q, the capacity $u(e_i, e_{i+1})$ is equal to the sum of λ_J over all active $J \in \mathcal{I}$ such that $J + e_i - e_{i+1} \in \mathcal{I}$. As all values in the series satisfy the property, so too do the capacities of each arc e_ie_{i+1}. Similarly, the capacity of se_1 is $x_{e_1} - y_{e_1}$ and

$$y_{e_1} = \sum_{J \in \mathcal{I}:\, \lambda_J > 0\ ,\ e_1 \in J} \lambda_J .$$

It follows that the augmentation amount ε satisfies the property. Consider the coefficients of the mutations of some set $J \in \mathcal{I}$. The coefficient of J^1 is

$$\lambda'_{J^1} = \max\left(0, \min\left(\lambda_J, L_J(e_1, f_1) - \varepsilon\right)\right).$$

Since 0, λ_J, and $L_J(e_1, f_1) - \varepsilon$ each satisfy the property, so too does λ'_{J^1}. Similarly, for $i > 1$, the coefficient of J^i is

$$\lambda'_{J^i} = \lambda_{J^i} + \max\left(0, \min\left(\lambda_J, L_J(e_i, f_i) - \varepsilon\right) - \sum_{1 \le j < i} \lambda'_{J^j} + \sum_{2 \le j < i} \lambda_{J^j}\right).$$

Once again, the quantities λ_{J^i}, 0, λ_J and $L_J(e_i, f_i) - \varepsilon$ each satisfy the property. So too does the quantity $\sum_{2 \le j < i} \lambda_{J^j}$. Since λ'_{J^1} satisfies the property, we may assume inductively that $\sum_{1 \le j < i} \lambda'_{J^j}$ does also, and hence the property is invariant under the action of the algorithm. Property (i) now follows immediately. $\square$

2.6 Basis Polyhedron Version

Cunningham's algorithm (Algorithm 2), and the presentation thereof, can be appreciably simplified by considering only vectors in the *basis polyhedron*

$$B_M = \left\{ y \in \mathbb{R}^S : y \ge 0,\ y(S) = r(S),\ y(A) \le r(A)\ \forall A \subseteq S \right\}$$

of M. Recall the min-max relation (2.1)

$$\max_{y \in P_M,\, y \le x} \{y(S)\} = \min_{A \subseteq S} \{r(A) + x(S \setminus A)\}.$$

Algorithm 2 works by iteratively raising y_e for some $e \in S$ such that $y_e < x_e$. An iteration does not affect the quantity y_f for any $f \ne e$. Starting from $y = 0$, the vector y is both always in the matroid polyhedron P_M and always less than or equal to x. When working over the basis polyhedron, the point y will certainly always be in the matroid polyhedron

P_M, but it might not be less than or equal to x. Consequently, the min-max relation above must be modified to

$$\max_{y \in B_M} \{y'(S)\} = \min_{A \subseteq S} \{r(A) + x(S \setminus A)\}, \tag{2.11}$$

where $y'_e = \min\{y_e, x_e\}$. Evidently, for any $A \subseteq S$,

$$y'(S) = y'(A) + y'(S \setminus A) \leq y(A) + x(S \setminus A) \leq r(A) + x(S \setminus A).$$

Moreover, given a maximizing vector $y \in P_M$ in equation (2.1), we can extend y to a basis vector $z \in B_M$. Then it follows easily that $z'(S) = y(S)$ and hence z is a maximizer in equation (2.11).

The algorithms here and in sections 4.3 and 4.4 actually solve the following more general problem: Given a matroid $M = (S, \mathcal{I})$ and a vector $x \in \mathbb{R}^S$, find a vector $y \in B_M$ such that $y \geq x$ or certify that none exists (that is, $x \notin P_M$). Evidently, any algorithm to solve this problem can also solve the matroid polyhedron membership problem.

Consider an st-dipath s,e_1,e_2,...,e_k,t in Algorithm 2. The crude augmentation can be thought of as having two parts. First, the augmentation raises y_{e_i} and lowers $y_{e_{i+1}}$ by some $\delta > 0$ for each i, $1 \leq i < k$. It does this by adding e_i to, and removing e_{i+1} from, some active $J \in \mathcal{I}$. In the final step, it adds e_k to some active $J \in \mathcal{I}$. The net effect is to raise y_{e_1} by δ without affecting any other entries in the vector y.

In the basis version, all independent sets in the combination are bases, hence all points y considered satisfy $y(S) = r(S)$. It follows that an augmentation cannot raise y_e for some $e \in S$ without also lowering y_f for some $f \in S$. In light of these observations, we formulate the auxiliary digraph $D = (N, A)$, where $N = S \cup \{s, t\}$ with the following arc-set.

- se for $e \in S$ with $x_e < y_e$.
- et for $e \in S$ with $x_e > y_e$.
- ef for $e, f \in S$ such that $f \in C(B, e)$ for some active B.

Let $Q = s$,e_1,e_2,...,e_k,t be an st-dipath in D. Modify the crude augmentation by simply omitting the final step; it is easy to see that the modified crude augmentation produces a valid convex combination of bases. Moreover, the modified crude augmentation raises

y_{e_1} and lowers y_{e_k} by δ. In addition to the bound on the value of δ given by the crude augmentation, δ must also be chosen to satisfy $\delta \leq \min\{y_{e_1} - x_{e_1}, x_{e_k} - y_{e_k}\}$. This rule ensures that no new arcs *se* or *et* are created during execution.

Conveniently, the grand augmentation is valid for the basis version. The only difference is that the grand augmentation should be performed with respect to the arcs e_1e_2, e_2e_3,...,$e_{k-1}e_k$, but not e_kt. The definition of the grand augmentation itself remains valid, however the effect is to raise y_{e_1} and lower y_{e_k} by ε (as opposed to just raising y_{e_1}). The proof of these facts is essentially identical to Theorem 2.3.1, and the bound on the number of active sets remains valid.

One of the benefits of the basis version is that, since any independent set considered is a basis, Lemma 2.4.3 is not necessary to prove that mutations are bases, nor to prove the running time bound. Moreover, as we observed in Section 2.4, the error in Cunningham's version of Lemma 2.4.2 (which appears as Lemma 4.3 in [3]) is only a problem when the lemma is applied to prove the analog of Lemma 2.4.3 (i.e. Lemma 4.4 in [3]). It is therefore unsurprising that the problems discussed in Section 2.4 and Section 2.5 vanish in the case of the basis version of the algorithm.

The proof of Lemma 2.5.1, which says that a critical arc of D is not an arc of the next auxiliary digraph in the sequence is very similar for the basis version. Recall that in [3] Cunningham proposed a lemma which states

For all $f \in S$, $d'(s, f) \geq d(s, f)$ and $d'(f, t) \geq d(f, t)$.

This proposition had to be modified to correct the algorithm. In the basis version, this assertion is true. To prove the statement, we require the following lemma, which appears as Lemma 4.6 in [3].

Lemma 2.6.1. *If ef is an arc of D' but not of D, then $e, f \in S$ and there exist vertices a, b of Q with a preceding b on Q such that ($a = f$ or) af is an arc of D and ($b = e$ or) eb is an arc of D.*

Proof. Suppose ef is an arc of D' but not D. By the choice of ε, $y'_f < x_f$ implies $y_f < x_f$ and $y'_e > x_e$ implies $y_e > x_e$. It follows that $e \neq s$ and $f \neq t$, so $e, f \in S$.

Since ef is an arc of D' but not D, there exists some mutation B' of some basis $B \in \mathcal{I}$ such that $f \in C(B',e) \setminus C(B,e)$ where $B' = B\Delta\{a_1, b_1, a_2, b_2, ..., a_\ell, b_\ell\}$ for some subsequence a_1b_1,a_2b_2,...,$a_\ell b_\ell$ of the arcs of Q. Therefore, we can apply Lemma 2.4.2 to e, f, B, B' and the arcs a_1b_1,a_2b_2,...,$a_\ell b_\ell$ (in the order that they appear on Q) to obtain a and b as asserted. □

This lemma now allows us to prove the monotonicity lemma.

Lemma 2.6.2. *For all $f \in S$, $d'(s,f) \geq d(s,f)$ and $d'(f,t) \geq d(f,t)$.*

Proof. Suppose $d'(s,f) < d(s,f)$, and choose f so that $d'(s,f)$ is minimized. By choosing the augmentation amount δ to satisfy $\delta \leq \min\{y_{e_1} - x_{e_1}, y_{e_k} - x_{e_k}\}$ in addition to the other constraints, no arcs sf or ft are created during execution. Therefore, $d'(s,f) > 1$, so there exists a predecessor $e \neq f$ on some shortest sf-dipath in D'. By the choice of f, $d'(s,e) \geq d(s,e)$, and so ef is an arc of D' but not D. Let $a, b \in S$ be as described by Lemma 2.6.1. Then

$$d'(s,f) = d'(s,e) + 1 \geq d(s,e) + 1 \geq d(s,b) = d(s,a) + 1 \geq d(s,f),$$

a contradiction. The proof that $d'(f,t) \geq d(f,t)$ is similar. □

The next lemma will be enough to bound the running time of the basis version of the algorithm. Recall that the st-dipath chosen at each iteration is the lexicographically least shortest st-dipath Q in D. Let P be the lexicographically least shortest st-dipath in D'.

Lemma 2.6.3. *If $d'(s,t) = d(s,t)$, then P is lexicographically greater than Q.*

Proof. Assume $d'(s,t) = d(s,t)$. Let $P = p_0, p_1, ..., p_k, p_{k+1}$ and $Q = q_0, q_1, ..., q_k, q_{k+1}$. Since some critical arc of Q is not an arc of D', P and Q are nonequal. Suppose P is lexicographically less than Q, and let i ($1 \leq i \leq k$) be the smallest index such that $p_i \neq q_i$; then $p_i < q_i$. It follows that $q_{i-1}p_i$ is an arc of D' but not D. By Lemma 2.6.1, ap_i is an arc of D for some vertex $a \neq q_{i-1}$ of Q. Since Q is chordless, it must be that $a = q_j$ for some $j \geq i$. But now, $b = q_\ell$ for some $\ell > j$ and $q_{i-1}q_\ell$ is a chord of Q, a contradiction. □

The remainder of the analysis now follows easily. There are $O(n)$ lengths of st-dipaths, and for a given length, there are $O(n^2)$ distinct st-dipaths, for a total of $O(n^3)$ grand augmentations. By Lemma 2.3.2, each such augmentation can be performed in $O(n^6 \cdot q)$ operations for a total of $O(n^9 \cdot q)$ operations. As in the previous section, the vector λ satisfies the property in Theorem 2.5.8. Similarly, the augmentation step can be extended to reduce the number of active bases to at most n before returning. This improves the running time of the algorithm to $O(n^6 \cdot q)$ but sacrifices the property in Theorem 2.5.8.

Chapter 3

Matroid Intersection

3.1 Introduction

The *matroid intersection problem* is defined as follows. Given two matroids $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ with rank functions r_1 and r_2 defined on a common ground set S, find a set $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ such that $|J|$ is maximized. The first question one might ask is, is there likely to be an algorithm to efficiently solve this problem? That is, can we succinctly certify that a set $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ is of maximum cardinality?

Let $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ and let $A \subseteq S$. Then

$$|J| = |J \cap A| + |J \cap (S \setminus A)| \leq r_1(A) + r_2(S \setminus A).$$

Therefore, if we can find $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ and $A \subseteq S$ such that $|J| = r_1(A) + r_2(S \setminus A)$, then we know that J is of maximum cardinality. The Matroid Intersection Theorem, due to Edmonds [7] states that, for any maximum cardinality $J \in \mathcal{I}_1 \cap \mathcal{I}_2$, there exists a set $A \subseteq S$ achieving equality.

Theorem 3.1.1. (Matroid Intersection Theorem) *For any two matroids $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ with rank functions r_1 and r_2,*

$$\max_{J \in \mathcal{I}_1 \cap \mathcal{I}_2} \{|J|\} = \min_{A \subseteq S} \{r_1(A) + r_2(S \setminus A)\}.$$

This theorem can be proved by induction. The proof we offer here will be in the form of an algorithm which terminates with a maximizing J and minimizing A.

The basic idea of the algorithm is to keep a set $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ whose cardinality is increased by one at each iteration. For each iteration, we construct an auxiliary digraph D, such that the existence of an st-dipath in D implies the existence of a larger common independent set. If there is no st-dipath, then we can use the information contained in the digraph to find a certificate.

We construct $D(M_1, M_2, J) = (N, A)$ as follows. Let $N = S \cup \{s, t\}$, where $s, t \notin S$ and let A contain the following arcs:

- se for $e \in S \setminus J$ with $J + e \in \mathcal{I}_1$
- fe for $e \in S \setminus J$, $f \in J$ with $f \in C_1(J, e)$
- ef for $e \in S \setminus J$, $f \in J$ with $f \in C_2(J, e)$
- et for $e \in S \setminus J$ with $J + e \in \mathcal{I}_2$

The following theorem shows that the auxiliary digraph has the desired properties.

Theorem 3.1.2. (Intersection Augmenting Path Theorem) *Let $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ be matroids, let $J \in \mathcal{I}_1 \cap \mathcal{I}_2$, and let D be the auxiliary digraph with respect to M_1, M_2, and J. Then there exists an st-dipath in D if and only if there is some $J' \in \mathcal{I}_1 \cap \mathcal{I}_2$ such that $|J'| > |J|$.*

Proof. First suppose that there exists no st-dipath in D. Let

$$A = \{a \in S : \text{there exists no } sa\text{−dipath in } D\}.$$

We claim that $J \cap A$ is an M_1-basis for A and $J \cap (S \setminus A)$ is an M_2-basis for $S \setminus A$. Suppose not; then there exists $e \in A \setminus J$ such that $(J \cap A) + e \in \mathcal{I}_1$. Since $J + e \notin \mathcal{I}_1$, we must have $f \in \mathrm{C}_1(J, e)$ for some $f \in J$, so fe is an arc of D. By the choice of A, $f \in A \cap J$. Similarly, for any element $f \neq e$ of $\mathrm{C}_1(J, e)$, there is an arc fe is an arc of D. Consequently, $\mathrm{C}_1(J, e) \subseteq (J \cap A) + e \notin \mathcal{I}_1$, a contradiction. The proof that $J \cap (S \setminus A)$ is an M_2-basis for $S \setminus A$ is similar. We can conclude that $|J| = r_1(A) + r_2(S \setminus A)$, and hence $|J|$ is maximized and A provides a certificate.

Conversely, suppose there exists an st-dipath in D, and let $s = e_0$, e_1, e_2,..., e_ℓ, $e_{\ell+1} = t$ be a chordless st-dipath. It is not hard to see that the sequence $e_\ell, e_{\ell-1}, ..., e_2$ satisfies the hypothesis of Proposition 2.4.1 with respect to M_1 and that the sequence $e_1, e_2, ..., e_{\ell-1}$ satisfies the hypothesis of Proposition 2.4.1 with respect to M_2. It follows that $J' = (J\Delta \{e_\ell, e_{\ell-1}, ..., e_2\}) + e_1 \in \mathcal{I}_1$ and $J' = (J\Delta \{e_1, e_2, ..., e_{\ell-1}\}) + e_\ell \in \mathcal{I}_2$. Therefore, $J' \in \mathcal{I}_1 \cap \mathcal{I}_2$ and $|J'| > |J|$, as required. □

The above discussion has established the correctness of Algorithm 3, again due to Edmonds [7].

Algorithm 3 Matroid Intersection Algorithm

```
1: $J \leftarrow \emptyset$, $D \leftarrow D(M_1, M_2, J)$
2: while there exists an $st$-dipath in $D$ do
3:     Let $s, e_1, e_2, ..., e_\ell, t$ be the vertex sequence of a chordless $st$-dipath
4:     $J \leftarrow J\Delta \{e_1, e_2, ..., e_\ell\}$
5:     $D \leftarrow D(M_1, M_2, J)$
6: end while
7: $A \leftarrow \{a \in S$ : there exists no $sa-$dipath in $D\}$
8: return $(J, A)$
```

Note that the algorithm also proves Theorem 3.1.1 since it always terminates with a $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ and $A \subseteq S$ for which $|J| = r_1(A) + r_2(S \setminus A)$.

The correctness having been established, it remains to determine the running time of the algorithm. We assume that the matroids have been given to us in the form of two independence oracles. Moreover, we assume that both independence oracles run in time $O(q)$.

Theorem 3.1.3. *Let $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ be matroids with rank functions r_1 and r_2 respectively. Then Algorithm 3, executed with M_1 and M_2 as inputs, returns a maximum cardinality set $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ and $A \subseteq S$ such that $|J| = r_1(A) + r_2(S \setminus A)$. Moreover, Algorithm 3 terminates in $O(n^3 \cdot q)$ operations.*

Proof. The Intersection Augmenting Path Theorem 3.1.2 verifies that the algorithm is correct. For the running time, since each iteration increases the cardinality of J by 1, the loop is executed $O(n)$ times. In each iteration, the algorithm must first find a chordless st-dipath, then perform the augmentation, then construct the auxiliary digraph. A chordless

st-dipath can be found by breadth-first search in $O(n^2)$ time, and the augmentation of J is accomplished in $O(n)$ time. To construct the arcs of the auxiliary digraph, the algorithm must make $O(n^2 \cdot q)$ tests of independence: two for each pair (e, f) with $f \in J$, $e \in S \setminus J$, and two for each $e \in S \setminus J$. Therefore, the algorithm runs in time $O(n^3 \cdot q)$. $\square$

Note that in our analysis, we have used breadth-first search to find a chordless st-dipath in D. In fact, breadth-first search determines a *shortest* st-dipath in D. As we shall see in the next section, the idea of choosing a shortest st-dipath, along with a creative analysis, can be combined to give an improved version of the algorithm.

3.2 Improved Matroid Intersection Algorithm

In this section, we study a modification of Algorithm 3 with an improved running time. The algorithm, due to Cunningham [5], is a generalization of the Hopcroft-Karp algorithm for maximum bipartite matching [15]. The problem of finding a maximum matching in a bipartite graph is a fundamental one, and is a special case of both network flows and matroid intersection in the following ways.

Let $G = (V, E)$ be a bipartite graph with vertex bipartition $V = X \cup Y$. By orienting all edges from X to Y, adding a supersource s, arcs sx for all $x \in X$, a supersink t, and arcs yt for all $y \in Y$, it is not hard to check that a maximum st-flow in the modified graph corresponds to a maximum matching in G. To formulate a matroid intersection problem, take E to be the common ground set. A set $A \subseteq E$ is independent in M_1 if no two edges in A meet at any vertex in X. Similarly, $A \subseteq E$ is independent in M_2 if no two edges of A meet at a vertex in Y. It is easy to check that M_1 and M_2 are matroids. Moreover, a common independent set of M_1 and M_2 is a subset of the edges of G, such that no two edges meet at any vertex $x \in X$ or any vertex $y \in Y$; that is, a common independent set is a matching.

Perhaps the simplest algorithm to compute a maximum matching in a bipartite graph is to iteratively search for an alternating path from an exposed vertex in X to an exposed vertex in Y. Since any maximum matching has size at most $|V|/2$, there are $O(|V|)$ iterations, and each augmenting path can be found in $O(|E|)$ time, giving a running time of $O(|V||E|)$.

In the analysis of the Edmonds-Karp algorithm, it is shown that there are $O(|V|)$ distinct lengths of augmenting paths, and for each length, there are $O(|E|)$ augmenting paths, each

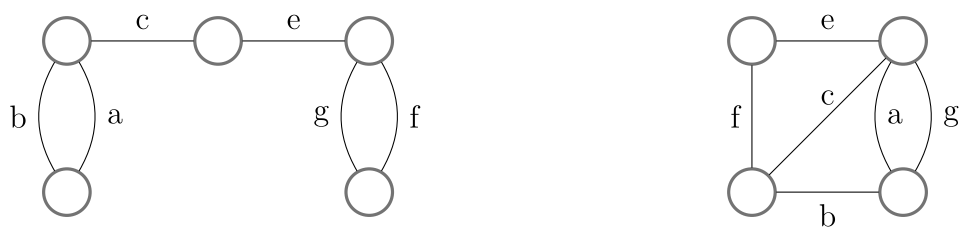

Figure 3.1: Counterexample to Theorem 4.2 of [5].

of which can be found in $O(|E|)$ time. The Hopcroft-Karp algorithm [15] borrows the idea of choosing a shortest augmenting path from the Edmonds-Karp algorithm. In their analysis, Hopcroft and Karp show that there are $O(|V|^{\frac{1}{2}})$ distinct lengths of augmenting paths, and that for a given length, all augmenting paths are disjoint. Consequently, all paths of a fixed length can be found in $O(|E|)$ time.

The proof of the running time of Cunningham's algorithm relies on the results that appear in [3] discussed in Section 2.4. In light of the problems with these results, the running time analysis in [5] needs to be modified. The fix presented here is along the lines of that found in Haselmayr's thesis [14]; as previously discussed, Haselmayr gave statements and proofs of Lemmas 2.4.2 and 2.4.3. Additionally, he gave statements and proofs of Lemmas 3.2.1 and 3.2.2.

At a high level, the algorithm works by constructing the auxiliary digraph and finding a collection of shortest disjoint *st*-dipaths. After using each path to perform an augmentation, it iterates. As we will show, there are $O(n^{\frac{1}{2}})$ iterations, and each can be performed in $O(n^2 \cdot q)$ time.

The first part of the analysis of the Hopcroft-Karp Algorithm shows that path lengths are nondecreasing. In the present case, we need the slightly stronger result Lemma 3.2.2. The theorem found in [5] (Theorem 4.2) simply states that for each $f \in S$, $d'(s, f) \geq d(s, f)$ and $d'(f, t) \geq d(f, t)$. In much the same way as we constructed a counterexample to the monotonicity lemma of [3] in Section 2.5, we can construct a counterexample to this theorem. The fact that counterexamples exist was first discovered by Haselmayr [14].

Consider the example in Figure 3.1. With respect to the graphic matroids of the graphs

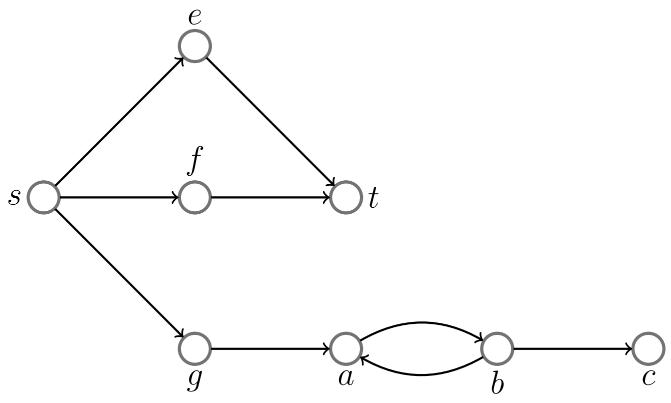

Figure 3.2: Auxiliary digraph with respect to M_1, M_2, and $J = \{a, c\}$.

above and the common independent set $\{a, c\}$, the auxiliary digraph is shown in Figure 3.2. Suppose we choose the st-dipath s, e, t and augment J to $J' = \{a, c, e\}$. Evidently, $J' + f \in \mathcal{I}_1$, so sf is an arc of the auxiliary digraph D' with respect to J'. Note also that $J' + f - c \in \mathcal{I}_2$, so fc is also an arc of D'. But now, $d'(s, c) < d(s, c)$, contradicting the monotonicity lemma (Theorem 4.2 of [5]). The problem is that the proof of this Theorem also relies on the incorrect lemmas of [3]. Using the results of Section 2.4, we can prove a modified version of the monotonicity lemma.

As in Section 2.5, we first use Lemmas 2.4.2 and 2.4.3 to prove necessary preconditions for the appearance of a new arc in the auxiliary digraph. We then go on to prove the monotonicity lemma. The statements and proofs of Lemmas 3.2.1 and 3.2.2 are due to Haselmayr [14].

Lemma 3.2.1. *Let D be the auxiliary digraph with respect to M_1, M_2 and J and let J' be obtained from J by an augmentation along some shortest st-dipath P in D. Let D' be the auxiliary digraph with respect to M_1, M_2 and J'. Suppose ef is an arc of D' but not of D. Then one of the following holds.*

(i) $e \in J'$ and sf is an arc of D,

(ii) $e \notin J'$ and et is an arc of D, or

(iii) there exist a and b of P with a preceding b such that ($a = f$ or af is an arc of D) and ($b = e$ or eb is an arc of D).

Proof. Assume that neither (i) nor (ii) holds. Let $P = s, e_1, e_2, ..., e_\ell, t$ and suppose $e \notin J'$. Since $ef \in A(D')$, we have $f \in J'$ and $J' + e - f \in \mathcal{I}_2$. As (ii) does not hold, either $e \in J$ or $J + e \notin \mathcal{I}_2$. In the latter case, $J + e - f \notin \mathcal{I}_2$ since $ef \notin A(D)$. Consider the following mapping:

e_1	e_2	...	$e_{\ell-2}$	$e_{\ell-1}$	e_ℓ
a_1	b_1	...	a_p	b_p	a_{p+1}

Clearly, $a_1, ..., a_p, a_{p+1} \notin J$ and $b_1, ..., b_p \in J$. With respect to M_2, and by the rules for the construction of the auxiliary digraph, conditions (i) and (iii) of Lemma 2.4.3 are satisfied. Since P is a shortest path, condition (ii) is also satisfied. Therefore, there exist k and ℓ with $1 \leq k \leq \ell \leq p$ such that $f \in C_2(J, a_k)$ and $b_\ell = e$ or $b_\ell \in C_2(J, e)$. Taking $a = a_k$ and $b = b_\ell$, a precedes b and we have that af is an arc of D and $b = e$ or eb is an arc of D, as required.

Now suppose $e \in J'$. Since $ef \in A(D')$, we must have $f \notin J'$ and $e \in C_1(J', f)$. As (i) does not hold, either $f \in J$ or $J + f \notin \mathcal{I}_1$. In the latter case, $J + f - e \notin \mathcal{I}_1$ since $ef \notin A(D)$. Consider the following mapping:

e_1	e_2	e_3	...	$e_{\ell-1}$	e_ℓ
a_{p+1}	b_p	a_p	...	b_1	a_1

Clearly, $a_1, ..., a_p, a_{p+1} \notin J$ and $b_1, ..., b_p \in J$. With respect to M_1, and by the rules for the construction of the auxiliary digraph, conditions (i) and (iii) of Lemma 2.4.3 are satisfied. Since the path chosen is shortest, it is also chordless, and hence condition (ii) is also satisfied. Therefore, there exist k and ℓ with $1 \leq k \leq \ell \leq p$ such that $e \in C_1(J, a_k)$ and $b_\ell = f$ or $b_\ell \in C_1(J, f)$ (note that the roles of e and f are the reverse of those found in Lemma 2.4.3). Taking $b = a_k$ and $a = b_\ell$, a precedes b and we have that $a = f$ or af is an arc of D and eb is an arc of D, as required. $\square$

Lemma 3.2.2. *Let D be the auxiliary digraph with respect to M_1, M_2 and J and let J' be obtained from J by an augmentation along some shortest st-dipath P in D. Let D' be the auxiliary digraph with respect to M_1, M_2 and J'. For each $f \in S$,*

(i) If $d(s, f) < d(s, t)$ then $d'(s, f) \geq d(s, f)$.

(ii) If $d(f, t) < d(s, t)$ then $d'(f, t) \geq d(f, t)$.

Proof. Suppose there exists $f \in S$ such that $d(s,f) < d(s,t)$ and $d'(s,f) < d(s,f)$. Subject to this, choose f so that $d'(s,f)$ is as small as possible. Evidently $f \neq s$, so we can choose the penultimate vertex e of some shortest sf-dipath in D'. By the choice of f, $d'(s,e) \geq d(s,e)$, and therefore ef is an arc of D' but not of D. By Lemma 3.2.1, there are three possibilities. As $1 \leq d'(s,f) < d(s,f)$, sf is not an arc of D, and hence (i) of Lemma 3.2.1 cannot be the case. Similarly, if (ii) holds, then

$$d(s,f) > d'(s,f) = d'(s,e) + 1 \geq d(s,e) + 1 \geq d(s,t),$$

contradicting that $d(s,f) < d(s,t)$. Therefore, we can find a and b of our shortest st-dipath in D as in (iii) of Lemma 3.2.1. As a precedes b on this path, $d(s,a) < d(s,b)$. However,

$$d(s,a) \geq d(s,f) - 1 \geq d'(s,f) = d'(s,e) + 1 \geq d(s,e) + 1 \geq d(s,b),$$

a contradiction. So (i) is proved, and the proof of (ii) is similar. □

Consider the algorithm $\mathcal{A}$, which is the modification of the Matroid Intersection Algorithm (Algorithm 3) that chooses a shortest st-dipath at each iteration. By Lemma 3.2.2, the sequence of lengths of the paths chosen is nondecreasing. We first show that $\mathcal{A}$ encounters $O(n^{\frac{1}{2}})$ distinct lengths of augmenting paths. The first two results serve to bound the length of a shortest st-dipath in the auxiliary digraph. The following Theorem is due to Cunningham [5], and the remainder of the analysis resembles that found in [5].

Theorem 3.2.3. *Let $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ be matroids, and let $J \in \mathcal{I}_1 \cap \mathcal{I}_2$. Let D be the auxiliary digraph with respect to M_1, M_2, and J, and let $J' \in \mathcal{I}_1 \cap \mathcal{I}_2$ with $|J'| > |J|$. Then there exist $|J'| - |J|$ pairwise disjoint st-dipaths in D, all of whose internal vertices are contained in $J \cup J'$.*

Proof. By Menger's Theorem (1.2.1), the maximum number of disjoint st-dipaths using only the vertices in $J \cup J'$ is equal to the minimum cardinality of a set $U \subseteq J \cup J'$ which separates s from t in the subgraph F of D induced by $J \cup J'$. Let T be the set of vertices reachable from s in $F - U$ and let $V = (J \cup J') \setminus (T \cup U)$.

As there is no arc se for $e \in V$, it must be the case that for all $e \in V \setminus J$, $J + e \notin \mathcal{I}_1$. Similarly, as there is no arc fe for $f \in T$ and $e \in V$, it follows that for all $f \in T \cap J$ and $e \in V \setminus J$, $f \notin \mathrm{C}_1(J,e)$. We claim that $J \cap (T \cup U)$ is an M_1-basis for $T \cup (U \cap J)$. Suppose not; then for some $e \in T \setminus J$, we have $J \cap (T \cup U) + e \in \mathcal{I}_1$. Since et is not an arc of D, $J + e \notin \mathcal{I}_2$. Therefore, there exists $f \in J \setminus (T \cup U) = J \cap V$ such that $f \in C_2(J,e)$, which implies that ef is an arc of D, contradicting the definition of V. Similarly, we can deduce that $J \cap (U \cup V)$ is an M_2-basis for $V \cup (U \cap J)$; if not, then there exists $f \in V \setminus J$ such that $J \cap (U \cup V) + f \in \mathcal{I}_2$. Since sf is not an arc of D, $J + f \notin \mathcal{I}_1$. Therefore, there exists $e \in T \cap J$ with $e \in C_1(J, f)$, which implies that ef is an arc of D, contradicting the definition of V.

Now,

$$\begin{aligned}
|J| + |J \cap U| &= |J| + |U| - |U \setminus J| \\
&= |J \cap (T \cup U)| + |J \cap (U \cup V)| \\
&\geq |J' \cap (T \cup (U \cap J))| + |J' \cap (V \cup (U \cap J))| \\
&= |J' \cap T| + |J' \cap U \cap J| + |J' \cap V| + |J' \cap U \cap J| \\
&\geq |J' \cap T| + |J' \cap V| + |U \cap J \cap J'|
\end{aligned}$$

By rearranging, we have

$$\begin{aligned}
|U| &\geq |J' \cap T| + |J' \cap V| + |J' \cap J \cap U| + |U \setminus J| - |J| \\
&= |J' \cap T| + |J' \cap V| + |J' \cap J \cap U| + |J' \cap U| - |J' \cap J \cap U| - |J| \\
&= |J' \cap T| + |J' \cap U| + |J' \cap V| - |J| \\
&= |J'| - |J|.
\end{aligned}$$

The assertion of the lemma follows. □

Lemma 3.2.4. *Let $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ be matroids, and let $J \in \mathcal{I}_1 \cap \mathcal{I}_2$. Let D be the auxiliary digraph with respect to M_1, M_2, and J, and suppose there exists an st-dipath in D. Let p be the cardinality of a maximum common independent set. Then there exists an st-dipath in D containing at most $|J|/(p - |J|)$ vertices from J.*

Proof. By Theorem 3.2.3, there exist at least $p - |J|$ disjoint st-dipaths in D. As the paths are disjoint, some path contains at most $\frac{|J|}{p-|J|}$ vertices from J, as required. □

During the course of execution, $\mathcal{A}$ generates a sequence $J_0, J_1, ..., J_p$ of common independent sets, starting with $J_0 = \emptyset$, and where each J_i ($1 \leq i \leq p$) is obtained from J_{i-1} by an augmentation along a shortest st-dipath P_{i-1} in the auxiliary digraph.

Lemma 3.2.5. *The number of distinct integers in the sequence $|P_0|, |P_1|, ..., |P_{p-1}|$ is at most $2\sqrt{p}$.*

Proof. Let $r = p - \lceil\sqrt{p}\rceil$ and let $V(P_r)$ be the vertex set of P_r. Then $|J_r| = r$ and by Lemma 3.2.4,

$$|V(P_r) \cap J_r| \leq \frac{|J_r|}{p - |J_r|} = \frac{p - \lceil\sqrt{p}\rceil}{\lceil\sqrt{p}\rceil} \leq \sqrt{p} - 1.$$

Moreover, as the subsequence $|P_0|, ..., |P_r|$ is nondecreasing, the same is true of P_i with respect to J_i for $0 \leq i \leq r$. By construction, the nodes of each st-dipath belong alternately to J and $S \setminus J$. Moreover, each st-dipath has even length. It follows that the subsequence contains at most $\sqrt{p}$ distinct integers. The subsequence $|P_{r+1}|, ..., |P_{p-1}|$ has $\lceil\sqrt{p}\rceil - 1 \leq \sqrt{p}$ elements, each of which may be distinct. Therefore, there are at most $2\sqrt{p}$ distinct integers in the sequence. $\square$

It follows from Lemma 3.2.2 and Lemma 3.2.5 that we can partition P_0, P_1, ..., P_{p-1} into $O(\sqrt{p})$ contiguous subsequences, where all paths in each subsequence have equal length. As in the previous chapter, the partition of the paths induces a partition of the augmentations performed, namely, part i consists of all augmentations performed on paths of length i. By Lemma 3.2.5, there are $O(\sqrt{p})$ stages in the algorithm. It remains to show that we can find and perform all augmentations for a stage in $O(n^2 \cdot q)$ time.

To that end, we define *layer* i for $i = 0$ to $d(s,t)$, as

$$L_i = \{f \in S \cup \{s,t\} : d(s,f) = i \text{ and } d(f,t) = d(s,t) - i\}.$$

Evidently, $f \in S$ belongs to some shortest st-dipath if and only if $f \in L_i$ for some i. Given M_1, M_2 and J, let $D = (N,A) = D(M_1, M_2, J)$. We can find $d(s,v)$ for each $v \in N$ using breadth-first search. Similarly, we can apply breadth-first search to the graph (N, A') where $A' = \{ef : fe \in A\}$ from source vertex t in order to determine $d(v,t)$ for each $v \in N$. Hence, we can construct the sets L_i for $i = 0$ to $d(s,t)$ in $O(n^2)$ time.

We are now ready to give Subroutine 4, which performs all augmentations for a stage. The input is assumed to be the matroids M_1 and M_2 together with a common independent set J. The subroutine is a variant of depth-first search. Given M_1, M_2 and J, let k be the length of a shortest st-dipath. The subroutine finds an st-dipath of length k and augments by it, then iterates. Termination is achieved when no st-dipath of length k remains (with respect to the augmented set J').

Subroutine 4 Stage Augmentation

```
 1: $(N, A) \leftarrow D(M_1, M_2, J)$
 2: construct the sets $L_i$ for $i = 0$ to $d(s, t)$
 3: $J' \leftarrow J$
 4: $U \leftarrow \emptyset$
 5: $p_u \leftarrow$ null for all $u \in N$
 6: $x \leftarrow s$
 7: $i \leftarrow 0$
 8: while $x \neq$ null do
 9:     if $x = t$ then
10:         follow $p$ to find a sequence $t = n_0, n_1, ..., n_{\ell+1} = s$
11:         $J' \leftarrow J' \Delta \{n_1, n_2, ..., n_\ell\}$
12:         $U \leftarrow U \cup \{n_1, n_2, ..., n_\ell\}$
13:         $x \leftarrow s$
14:         $i \leftarrow 0$
15:     else
16:         $i \leftarrow$layer of $x$
17:         if there exists $y \in L_{i+1} \setminus U$ with $xy$ an arc of $D(M_1, M_2, J')$ then
18:             $p_y \leftarrow x$
19:             $x \leftarrow y$
20:             $i \leftarrow i + 1$
21:         else
22:             $U \leftarrow U \cup \{x\}$
23:             $x \leftarrow p_x$
24:             $i \leftarrow i - 1$
25:         end if
26:     end if
27: end while
28: return $J'$
```

The subroutine keeps a set $U \subseteq N$, whose vertices are said to be useless. Formally, within a stage where the length of a shortest st-dipath is k, we label a vertex $u \in N$ *useless* if it is known to belong to no st-dipath (in the digraph with respect to J') of length k. Notice that the algorithm places an element x in U under one of two conditions: first, if it belongs to some st-dipath which was already used (and is not s or t), and second, if it belongs to layer i and there is no arc xy of $D(M_1, M_2, J')$ for some nonuseless y in layer $i+1$. In the latter case, it is easy to see that x is useless. In the former, suppose $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ is augmented to J' using a shortest st-dipath which contains $f \in S$. Then either $f \in J \setminus J'$ or $f \in J' \setminus J$. In either case, Lemma 3.2.2 implies that we have both that $d'(s,f) > d(s,f)$ and $d'(f,t) > d(f,t)$, since the parity of the length of each such path changes. It follows that each $f \in S$ belongs to at most one path per stage.

On line 17, the subroutine seeks an element $y \in L_{i+1} \setminus U$ such that xy is an arc of $D(M_1, M_2, J')$. It is important to note that in this step, the subroutine does not actually construct the auxiliary digraph $D(M_1, M_2, J')$, but instead uses a single test of independence to determine if the arc would be present. This is actually necessary to achieve the improved running time; if it did construct the auxiliary digraph for each augmentation, then there would be $O(n)$ constructions of the auxiliary digraph, each requiring $O(n^2 \cdot q)$ operations, giving a $O(n^3 \cdot q)$ running time bound. If instead, the algorithm constructs the auxiliary digraph just once for each distinct length of augmenting path, then the $O(n^{\frac{5}{2}} \cdot q)$ bound is attainable.

Lemma 3.2.6. *Let $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ be matroids, and let $J \in \mathcal{I}_1 \cap \mathcal{I}_2$. Let $D = D(M_1, M_2, J)$, $d(s,t) = k$, and let J' be the output of Subroutine 4 with M_1, M_2 and J as inputs. Then there is no st-dipath of length k in $D(M_1, M_2, J')$. Moreover, where p is the maximum cardinality of a common independent set and $O(q)$ is the complexity of the independence oracles for M_1 and M_2, Subroutine 4 terminates in $O(p \cdot n \cdot q)$ time.*

Proof. Notice that when the algorithm visits a vertex x, there are only two possible outcomes: either the algorithm will find an st-dipath travelling through the vertex, or it will attempt to find a path through each valid neighbour of x unsuccessfully. In either case, x is marked as useless. It follows that each neighbour of s (in L_1) will eventually be marked useless, which implies that s itself will eventually be marked useless. Therefore, the algorithm terminates, since upon marking s as useless, it sets $x =$ null.

As discussed above, the algorithm marks a vertex as useless only if it is contained in no shortest st-dipath with respect to the current common independent set. By the path length monotonicity property (Lemma 3.2.2), this vertex cannot belong to any shortest st-dipath for the rest of the stage. Since at termination, $s \in U$, we can be certain that there exists no st-dipath in $D(M_1, M_2, J')$ of length k, which implies that the stage is over.

To determine the running time of the subroutine, note that the complexity of the while loop is dominated by the number of tests for arcs xy where $y \in L_{i+1} \setminus U$. But since the algorithm tests for the existence of each such arc at most once, it performs $O(p \cdot n)$ such tests. Since each test requires a test of independence, the running time is $O(p \cdot n \cdot q)$, as required. □

Algorithm 5 Improved Matroid Intersection Algorithm

```
$J \leftarrow \emptyset$
**repeat**
    $J' \leftarrow J$
    $J \leftarrow$ stage augmentation $(M_1, M_2, J)$
**until** $J' = J$
$D \leftarrow D(M_1, M_2, J)$
$A \leftarrow \{e \in A :$ there exists no $se-$dipath in $D\}$
**return** $(J, A)$
```

The Improved Matroid Intersection Algorithm is shown as Algorithm 5.

Theorem 3.2.7. *Given two matroids M_1 and M_2 as input, Algorithm 5 returns $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ and $A \subseteq S$ such that $|J| = r_1(A) + r_2(S \setminus A)$. Moreover, Algorithm 5 terminates in $O(n^{\frac{5}{2}} \cdot q)$ operations.*

Proof. We first show that $J \in \mathcal{I}_1 \cap \mathcal{I}_2$. For this, it suffices to show that any st-dipath along which an augmentation is performed is chordless. Since Subroutine 4 always begins by augmenting along a shortest st-dipath, it follows inductively from Lemma 3.2.2 that all st-dipaths chosen are chordless.

When the algorithm breaks the loop, it does so because no st-dipaths were found in the auxiliary digraph. It follows from the Intersection Augmenting Path Theorem 3.1.2 that J is a maximum cardinality common independent set and A certifies this. For the running time, by Lemma 3.2.5, there are $O(n^{\frac{1}{2}})$ iterations of the loop. In each iteration, we call the stage augmentation algorithm (Subroutine 4), which by Lemma 3.2.6, terminates in

$O(n^2 \cdot q)$ operations. In each iteration, we also construct the auxiliary digraph, which as before, can be constructed in $O(n^2 \cdot q)$ operations, so Algorithm 5 terminates in $O(n^{\frac{5}{2}} \cdot q)$ operations, as required. $\square$

Chapter 4

Submodular Function Minimization

4.1 Introduction

Let S be a finite set and let $g : 2^S \to \mathbb{R}$; we call g *submodular* if, for all $A, B \subseteq S$, we have $g(A) + g(B) \geq g(A \cup B) + g(A \cap B)$. The *submodular function minimization problem* is, given a submodular function g defined on a set S, to find a subset A of S such that $g(A)$ is minimized. Since any set minimizing $g'(A) = g(A) - g(\emptyset)$ also minimizes g, we assume that $g(\emptyset) = 0$. Moreover, we assume that g is given in the form of a *value-giving oracle*, that is, a function which, given $A \subseteq S$, returns the value $g(A)$.

Submodular function minimization contains both matroid polyhedron membership and the decision version of matroid intersection as special cases in the following ways. Given an instance $M = (S, \mathcal{I})$ and $x \in \mathbb{R}^S$ of matroid polyhedron membership, define a function $g(A) = r(A) - x(A)$. It is easy to see that g is submodular. Moreover, if the minimum of $g(A)$ over all $A \subseteq S$ is non-negative, then $x(A) \leq r(A)$ for all $A \subseteq S$, that is, $x \in P_M$. Similarly, given two matroids $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$, define a function $g(A) = r_1(A) + r_2(S \setminus A)$. Then g is easily seen to be submodular and, by the Matroid Intersection Theorem 3.1.1, a minimizer of g determines the maximum cardinality of a common independent set.

The *submodular polyhedron* associated with g is defined by

$$P_g = \left\{ x \in \mathbb{R}^S : x(A) \leq g(A) \ \forall A \subseteq S \right\}.$$

In this chapter, algorithms for minimizing a submodular function g will work over the *base polyhedron* of g, defined by

$$B_g = \left\{ x \in \mathbb{R}^S : x(S) = g(S),\ x(A) \le g(A)\ \forall A \subseteq S \right\}.$$

The min-max theorem characterizing a minimizer of g is a special case of the Polymatroid Intersection Theorem of Edmonds [7].

Theorem 4.1.1. (Submodular Min-Max Theorem) *Let $g : 2^S \to \mathbb{R}$ be a submodular function. Then*

$$\max_{y \in B_g} \left\{ y^-(S) \right\} = \min_{A \subseteq S} \left\{ g(A) \right\} \tag{4.1}$$

where the vector y^- is defined by $y^-_e = \min\{0, y_e\}$.

The proof of this theorem relies on the following two lemmas.

Lemma 4.1.2. *Let $A, B \subseteq S$ be such that $y(A) = g(A)$ and $y(B) = g(B)$. Then $y(A \cup B) = g(A \cup B)$ and $y(A \cap B) = g(A \cap B)$.*

Proof. Identical to the proof of Lemma 2.1.3. □

Lemma 4.1.3. *If y is a maximizer in Theorem 4.1.1, then for each pair $u, v \in S$ with $y_u < 0$ and $y_v > 0$, there exists some $A \subseteq S$ with $y(A) = g(A)$ such that $u \in A$ but $v \notin A$.*

Proof. Suppose not. Then there exists a pair $u, v \in S$ with $y_u < 0$ and $y_v > 0$ and such that every $U \subseteq S$ with $u \in U$ and $v \notin U$ has $y(U) < g(U)$. Consider the vector $y' = y + \varepsilon(\chi_u - \chi_v)$ for some sufficiently small $\varepsilon > 0$. Observe that the quantity $y'(U)$ differs from $y(U)$ only if precisely one of u or v is contained in U. If $v \in U$ and $u \notin U$, then $y'(U) < y(U) \le g(U)$. If $u \in U$ and $v \notin U$ then $y'(U) > y(U)$, so $y'(U) \le g(U)$ is satisfied only if ε satisfies

$$\varepsilon \le \min_{U \subseteq S : u \in U,\ v \notin U} \left\{ g(U) - y(U) \right\}.$$

If in addition, ε is chosen to satisfy $\varepsilon \le \min\{y_v, -y_u\}$, then we also have that $y'_v = y_v - \varepsilon \ge 0$ and $y'_u = y_u + \varepsilon \le 0$. It follows that y is not a maximizer, a contradiction. □

Proof. (Of the Submodular Min-Max Theorem 4.1.1) Let y be a maximizing vector in (4.1), and let $X \subseteq S$. Then

$$y^-(S) = y^-(X) + y^-(S \setminus X) \le y(X) + y^-(S \setminus X) \le g(X).$$

So the maximum in (4.1) is always less than or equal to the minimum. It remains to show that equality holds. By Lemma 4.1.3, for each $u \in S$ with $y_u < 0$, and each $v \in S$ with $y_v > 0$, there exists a set $A^{u,v}$ such that $y(A^{u,v}) = g(A^{u,v})$. For each $u \in S$ with $y_u < 0$, define A^u to be the intersection of the sets $A^{u,v}$ for each $v \in S$ with $y_v > 0$. Let A be the union of A^u for all such u. By Lemma 4.1.2, the set A satisfies $y(A) = g(A)$. Since $e \notin A^u$ for any $e \in S$ with $y_e > 0$, $e \notin A$. Moreover, for each $e \in S$ with $y_e < 0$, $e \in A$. It follows that $g(A) = y(A) = y^-(A) = y^-(S)$, as required. □

The first algorithm to solve the submodular minimization problem was the ellipsoid method [17], which was shown by Grötschel, Lovász and Schrijver [13] to be applicable to rational-valued submodular functions. However, their running time bound is not strongly polynomial. Based in part on the algorithm seen in Chapter 2, Cunningham gave a combinatorial algorithm for minimizing submodular functions in 1985 [4]; however the running time of this algorithm is not polynomial. It was not until 2000 that Schrijver [21], and Fleischer, Fujishige and Iwata [16] gave combinatorial and strongly polynomial time algorithms for minimizing a submodular function.

Section 2 of this chapter presents Schrijver's algorithm. Section 3 considers the behaviour of Schrijver's algorithm in the context of the matroid polyhedron membership problem and refines Schrijver's algorithm in this special case to give another algorithm for solving the problem. Section 4 presents a push-relabel algorithm for the matroid polyhedron membership problem, due to Frank and Miklós. Finally, section 5 applies the push-relabel framework to Schrijver's algorithm to obtain a push-relabel algorithm for submodular function minimization.

4.2 Schrijver's Algorithm

Schrijver's algorithm [21] builds on Cunningham's algorithm for submodular minimization [4], which in turn builds on the matroid polyhedron membership algorithm [3]. Cunningham's submodular minimization algorithm also relies on a subroutine of Bixby, Cunningham, and Topkis [1]. Given an extreme point v of the submodular polyhedron, the subroutine returns a partial ordering which determines nearby extreme points of the form $v + \varepsilon(\chi_e - \chi_f)$ or $v + \varepsilon\chi_e$ with $\varepsilon > 0$. Cunningham's algorithm keeps a point y in the submodular polyhedron as a convex combination of extreme points. The partial orders returned by the subroutine are then used to construct an auxiliary digraph, and augmentations are similar to augmentations in the matroid polyhedron membership algorithm.

The augmentation step described by Cunningham in [4] can be likened to the crude augmentation of Chapter 2. In particular, after augmenting along an *st*-dipath, every arc of that path may occur in the next iteration. Recall that in Chapter 2, this problem was remedied by designing the grand augmentation, which provably reduces to zero the capacity of at least one arc of the dipath used; this provides the basis for the polynomial bound on the algorithm. Unfortunately, no analog to the grand augmentation is defined. Instead, assuming the input function is integer-valued, Cunningham proves that the augmentation amount δ can be chosen large enough to give a pseudo-polynomial bound on the running time of the algorithm.

Given the similarities to the matroid polyhedron membership algorithm, it seems that there should exist an analog to the grand augmentation. Such an analog may exist, but was not described by Cunningham in [4], nor in the years between 1985 and 2000 while the problem of submodular minimization remained open. In section 2.6, we saw how the algorithm for matroid polyhedron membership simplified considerably when modified to work over the basis polyhedron of M rather than the entire matroid polyhedron. Schrijver's algorithm takes this approach.

The principal difference between the algorithms is that Schrijver's algorithm performs small, local changes. The algorithm of Section 2.6 looks for a sequence $e_1, e_2, ..., e_k$ of elements of S such that $y_{e_1} < x_{e_1}$ and $y_{e_k} > x_{e_k}$ then raises y_{e_1} and lowers y_{e_k} by $\varepsilon > 0$. Schrijver's augmentation step instead works on a single arc, but by a complicated selection rule for the elements, a similar effect is achieved. This has the advantage of being much simpler to explain and prove.

Let S be a finite set and $g : 2^S \to \mathbb{R}$ be a submodular function. Let $\prec$ be an ordering on S. For each $e \in S$, let $e_\prec = \{a \in S : a \prec e\}$. Define a vector $b^\prec \in \mathbb{R}^S$ by

$$b_e^\prec = g(e_\prec + e) - g(e_\prec).$$

We call b a *base* of g.

Proposition 4.2.1. *Let g be a submodular function defined on S. Then for any order $\prec$ on the elements of S, the vector $b^{\prec}$ is an element of B_g.*

Proof. We first show that $b^{\prec}(A) \leq g(A)$ for all $A \subseteq S$, by induction on $|A|$. For $|A| = 0$, the assertion is trivially true, so assume $|A| > 0$. Let e be the greatest element (according to $\prec$) of A. By the submodularity of g,

$$g(A) \geq g(A \cup e_{\prec}) + g(A \cap e_{\prec}) - g(e_{\prec}).$$

By the choice of e, $A \cap e_{\prec} = A - e$ and $A \cup e_{\prec} = e_{\prec} + e$. Now,

$$\begin{aligned} g(A) &\geq g(e_{\prec} + e) + g(A - e) - g(e_{\prec}) \\ &\geq g(e_{\prec} + e) - g(e_{\prec}) + b^{\prec}(A - e) \\ &= b^{\prec}_e + b^{\prec}(A - e) \\ &= b^{\prec}(A). \end{aligned}$$

It remains to show that $b^{\prec}(S) = g(S)$. We have

$$\begin{aligned} b^{\prec}(S) &= \sum_{e \in S} b^{\prec}_e \\ &= \sum_{e \in S} g(e_{\prec} + e) - g(e_{\prec}) \\ &= g(S) - g(\emptyset) \\ &= g(S), \end{aligned}$$

as required. □

The basic idea of the algorithm is to keep a point $y \in B_g$ which is represented as a convex combination of bases of g. As in the previous algorithms we have seen, this is accomplished by means of an augmenting path scheme. For each ordering $\prec_i$ on S, we maintain a coefficient λ^i of the base generated by $\prec_i$. We call an ordering $\prec_i$ (or equivalently, the base generated by that ordering) *active* if $\lambda^i > 0$. The algorithm begins with an arbitrary ordering $\prec_1$ on S and $\lambda^1 = 1$.

Given all active orderings $\prec_1, \prec_2, ..., \prec_k$, the auxiliary digraph D has vertex set S and arc set $\{uv : u \prec_i v \text{ for some } 1 \leq i \leq k\}$. Let $y = \sum_{1 \leq i \leq k} \lambda^i b^{\prec_i}$ and define $P = \{e \in S : y_e > 0\}$ and $N = \{e \in S : y_e < 0\}$.

Theorem 4.2.2. *If D contains no (P, N)-dipath, then the set*

$$A = \{e \in S : \textit{there exists an } (e, N) - \textit{dipath in } D\}$$

and the vector y are an optimal pair in the Submodular Min-Max Theorem (4.1.1).

Proof. Evidently, $P \cap A = \emptyset$, so for each $e \in A$, $y_e \leq 0$. It follows that $y(A) \leq y(R)$ for all $R \subseteq S$. Moreover, $N \subseteq A$, so $y(A) = y^-(S)$.

Consider the sum

$$\begin{aligned} b^{\prec_i}(A) &= \sum_{e \in R} b_e^{\prec_i} \\ &= \sum_{e \in R} g(e_{\prec_i} + e) - g(e_{\prec_i}). \end{aligned}$$

Note that, if $e \in A$ and $a \prec_i e$, then $a \in A$. It follows that the above series is telescoping, and equal to $g(A)$. Now,

$$\begin{aligned} y(A) &= y^-(S) \\ &= \sum_{e \in A} \sum_{i=1}^{k} \lambda^i b_e^{\prec_i} \\ &= \sum_{i=1}^{k} \lambda_i \sum_{e \in A} b_e^{\prec_i} \\ &= \sum_{i=1}^{k} \lambda^i b{\prec_i}(A) \\ &= g(A), \end{aligned}$$

as required. □

If there is a (P, N)-dipath in D, then there exists a vector $y_1 \in B_g$ such that $y^-(S) < y_1^-(S)$. In what follows, we describe Schrijver's augmentation step for the case that there exists a (P, N)-dipath in D, and prove that repeated application of the augmentation step yields a polynomial-time algorithm for submodular minimization.

As stated, Schrijver's algorithm is slightly different in that augmentations are performed on a single arc of the path, and with respect to a single active ordering. Ultimately, the algorithm seeks to raise y_q for elements $q \in N$ and lower y_p for elements $p \in P$ so that $N = \emptyset$ or so that there is no (P, N)-dipath. Then by Theorem 4.1.1 or Theorem 4.2.2, y is a maximizer. At a high level, the algorithm works as follows. It constructs the auxiliary digraph and uses a consistent selection rule to select the last arc uv of a shortest (P, N)-dipath. It chooses v so that $d(P, v)$ is greatest among all $v \in N$. Then the augmentation step raises y_v and lowers y_u by some $\delta \geq 0$. In the $\delta = 0$ case, the algorithm does some rearranging. In the $\delta > 0$ case, if $u \notin P \cup N$, then u enters N. If $u \in P$ and $y_u \geq \delta$, then u does not enter N. The idea is that, since a shortest path from some element of P to u is shorter than the shortest path to v, the quantity δ, which should eventually be absorbed by elements of P, travels 'closer' to P with each iteration.

Let $\prec$ be an ordering on S and let $u, v \in S$ with $u \prec v$. Define the interval

$$(u, v]_\prec = \{e \in S : u \prec e \preceq v\}.$$

Define the open interval $(u, v)_\prec$, the half-open interval $[u, v)_\prec$ and the closed interval $[u, v]_\prec$ similarly. Furthermore, define $\prec^{u,v}$ to be the ordering on S obtained from $\prec$ by setting $v \prec e$ for each $e \in [u, v)_\prec$. The following lemma is helpful in describing and proving the augmentation step.

Lemma 4.2.3. *For each $e \in S$,*

$$\begin{aligned} b_e^{\prec^{u,v}} &\leq b_e^\prec && \text{if } u \preceq e \prec v \\ b_e^{\prec^{u,v}} &\geq b_e^\prec && \text{if } e = v \\ b_e^{\prec^{u,v}} &= b_e^\prec && \text{otherwise} \end{aligned}$$

Proof. First suppose $u \preceq e \prec v$. Consider the sets $e_\prec + e$ and $e_{\prec^{u,v}} = e_\prec + v$. These sets have union and intersection $e_{\prec^{u,v}} + e$ and $e_\prec$, respectively. It follows from the submodularity of g that

$$g(e_{\prec^{u,v}} + e) - g(e_{\prec^{u,v}}) \leq g(e_\prec + e) - g(e_\prec),$$

or equivalently, that $b_e^{\prec^{u,v}} \leq b_e^\prec$.

Now suppose $e = v$. Consider the sets $e_{\prec^{u,v}} + e$ and $e_\prec$. These sets have union and intersection $e_\prec + e$ and $e_{\prec^{u,v}}$, respectively. It follows from the submodularity of g that

$$g(e_{\prec^{u,v}} + e) - g(e_{\prec^{u,v}}) \geq g(e_\prec + e) - g(e_\prec),$$

or equivalently, that $b_e^{\prec^{u,v}} \geq b_e^{\prec}$.

Finally, suppose neither of the above cases holds. Then $e_{\prec^{u,v}} = e_{\prec}$ and hence $b_e^{\prec^{u,v}} = b_e^{\prec}$. □

The idea of the augmentation step is to work on an arc uv of the auxiliary digraph with $v \in N$. The step consists of choosing an active order $\prec$ for which $u \prec v$ and computing a representation for the vector $b^{\prec} + \delta(\chi_v - \chi_u)$ as a convex combination of the vectors $b^{\prec^{u,e}}$ for $e \in (u, v]_{\prec}$, and where $\delta \geq 0$. Equivalently, we show how to express the vector $\delta(\chi_v - \chi_u)$ as a convex combination of the vectors $b^{\prec^{u,e}} - b^{\prec}$ for $e \in (u, v]_{\prec}$.

Consider the vectors $b^{\prec^{u,e}} - b^{\prec}$ for $e \in (u, v]_{\prec}$. By Lemma 4.2.3, each such vector has value 0 for any element of S not contained in the interval $[u, v]_{\prec}$. Lemma 4.2.3 also implies that, on the interval $[u, v]_{\prec}$, the vectors $b^{\prec^{u,e}} - b^{\prec}$ have the form

$$\begin{array}{c} \\ u \\ e_1 \\ e_2 \\ e_3 \\ \cdots \\ e_k \\ v \end{array}
\begin{array}{c} \prec^{u,e_1} \\ \left[\begin{array}{c} - \\ + \\ 0 \\ 0 \\ \cdots \\ 0 \\ 0 \end{array}\right] \end{array},
\begin{array}{c} \prec^{u,e_2} \\ \left[\begin{array}{c} - \\ - \\ + \\ 0 \\ \cdots \\ 0 \\ 0 \end{array}\right] \end{array},
\begin{array}{c} \prec^{u,e_3} \\ \left[\begin{array}{c} - \\ - \\ - \\ + \\ \cdots \\ 0 \\ 0 \end{array}\right] \end{array}, \ldots,
\begin{array}{c} \prec^{u,v} \\ \left[\begin{array}{c} - \\ - \\ - \\ - \\ \cdots \\ - \\ + \end{array}\right] \end{array}$$

The portions of the vectors corresponding to elements of S coming before u or after v (according to $\prec$) are not shown above. The first entry in each vector corresponds to u and the last corresponds to v. The first vector in the list corresponds to the order obtained by exchanging u and its successor, and the last vector corresponds to the order $b^{\prec^{u,v}}$. A $+$ entry corresponds to a value which is greater than or equal to 0, and a $-$ entry corresponds to a value which is less than or equal to 0.

By the definition of $b^{\prec}$, we have $b^{\prec^{u,e}}(S) = g(S) = b^{\prec}(S)$. It follows that each vector in the sequence sums to 0. For the augmentation step, we consider two cases. In the first case, there exists some $e \in (u, v]_{\prec}$ such that $b_e^{\prec^{u,e}} = 0$. In terms of the vectors above, this corresponds to a vector whose $+$ entry is equal to 0. It follows that $b^{\prec^{u,e}} - b^{\prec} = 0$. In this case, we choose $\delta = 0$ and the representation $b^{\prec} = b^{\prec^{u,e}}$.

Otherwise, each + entry is strictly positive. In this case, we can convert the sequence of vectors above into a sequence of the form

$$\begin{array}{c} \\ u \\ e_1 \\ e_2 \\ e_3 \\ \dots \\ e_k \\ v \end{array}
\begin{array}{c} \prec^{u,e_1} \\ \begin{bmatrix} - \\ + \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \end{bmatrix} \end{array},
\begin{array}{c} \prec^{u,e_2} \\ \begin{bmatrix} - \\ 0 \\ + \\ 0 \\ \dots \\ 0 \\ 0 \end{bmatrix} \end{array},
\begin{array}{c} \prec^{u,e_3} \\ \begin{bmatrix} - \\ 0 \\ 0 \\ + \\ \dots \\ 0 \\ 0 \end{bmatrix} \end{array}, \dots,
\begin{array}{c} \prec^{u,v} \\ \begin{bmatrix} - \\ 0 \\ 0 \\ 0 \\ \dots \\ 0 \\ + \end{bmatrix} \end{array}$$

By the sign pattern, each modified vector in the sequence can be represented as a non-negative linear combination of the vectors $b^{\prec^{u,e}} - b^{\prec}$ for $e \in (u,v]_{\prec}$. Moreover, each modified vector has component sum 0. In particular, this applies to the vector $[-\ 0\ 0\ 0\ 0\ \dots\ 0\ +]^{\top}$, which is already equal to $\Delta(\chi_v - \chi_u)$ for some positive Δ. It follows that by scaling, we can represent the vector $\delta(\chi_v - \chi_u)$ as a convex combination of the vectors $b^{\prec^{u,e}} - b^{\prec}$ for $e \in (u,v]_{\prec}$ and some positive δ. Schrijver's subroutine is outlined in Subroutine 6. The subroutine requires a pair $u, v \in S$ and an ordering $\prec$ such that $u \prec v$. It returns a vector $\kappa \geq 0$ parameterized by the elements of $(u,v]_{\prec}$ such that

$$\sum_{e \in (u,v]_{\prec}} \kappa_e = 1 \quad \text{and}$$

$$\sum_{e \in (u,v]_{\prec}} \kappa_e \cdot b^{\prec^{u,e}} = b^{\prec} + \delta(\chi_v - \chi_u).$$

We have seen how to construct the auxiliary digraph, how to find a certificate if no (P,N)-dipath exists, and how to perform an augmentation when one does. The last part of the algorithm is the choice of an arc for augmentation. For this part, we require a fixed ordering on S, say $S = \{1, 2, ..., n\}$ with the usual order. Schrijver's algorithm is outlined in Algorithm 7.

Let $\prec_1, \prec_2, ..., \prec_k$ be the set of active orderings and let $\alpha = \max_{1 \leq i \leq k} \{|(u,v]_{\prec_i}|\}$. Assume, without loss of generality, that $|(u,v]_{\prec_1}| = \alpha$. Using the augmentation step, we can represent the vector $b^{\prec_1} + \delta(\chi_v - \chi_u)$ as a convex combination of the vectors $b^{\prec_1^{u,e}}$ for $e \in (u,v]_{\prec_1}$. Substituting this into the equation for y (the current element of B_g), we obtain a representation for the vector $y' = y + \lambda^1 \delta(\chi_v - \chi_u)$ as a convex combination of the vectors $b^{\prec_i}$

Algorithm 6 Schrijver's Subroutine

```
for each $e \in (u,v]_\prec$ do
    compute $b^{\prec^{u,e}}$
end for
if there exists $e \in (u,v]_\prec$ with $b^{\prec^{u,e}} = b^\prec$ then
    $\kappa_e \leftarrow 1$
else
    $k_v \leftarrow 1$
    $b' \leftarrow b^{\prec^{u,v}} - b^\prec$
    for each $e \in (u,v)_\prec$ (in reverse order) do
        $k_e \leftarrow \frac{b^\prec_e - b'_e}{b^{\prec^{u,e}}_e - b^\prec_e}$
        $b' \leftarrow b' + k_e(b^{\prec^{u,e}} - b^\prec)$
    end for
    $\ell \leftarrow \sum_{e \in (u,v]_\prec} k_e$
    for each $e \in (u,v]_\prec$ do
        $\kappa_e \leftarrow \frac{k_e}{\ell}$
    end for
end if
return $\kappa$
```

Algorithm 7 Schrijver's Submodular Function Minimization Algorithm

```
$\prec_1 \leftarrow$ arbitrary order on $S$
$\lambda^1 \leftarrow 1$
$D \leftarrow$ auxiliary digraph with respect to $y = b^{\prec_1}$
**while** there exists a $(P, N)$-dipath in $D$ **do**
    $v \leftarrow$ greatest element of $N$ with $d(P, v) < \infty$ maximum
    $u \leftarrow$ greatest element of $S$ with $d(P, u) = d(P, v) - 1$ such that $u \prec v$ for active $\prec$
    $\prec_i \leftarrow$ any active ordering maximizing $|(u, v]_{\prec_i}|$
    $\kappa \leftarrow$ result of Subroutine 6 with $u, v$ and $\prec_i$ as inputs
    $y' \leftarrow \sum_{j \neq i} \lambda^j b^{\prec_j} + \lambda^i \sum_{e \in (u,v]_{\prec_i}} \kappa_e b^{\prec_i^{u,e}}$
    **if** $y'_v > 0$ **then**
        $y \leftarrow$ point $z$ on line segment $yy'$ with $z_v = 0$
    **else**
        $y \leftarrow y'$
    **end if**
    reduce number of active bases in convex combination for $y$
    $D \leftarrow$ auxiliary digraph with respect to $y$
**end while**
$A \leftarrow \{e \in S : \text{there exists an } (e, N) - \text{dipath in } D\}$
**return** $(y, A)$
```

$(i = 2, ..., k)$ and $b^{\prec_1^{u,e}}$ $(e \in (u, v]_{\prec_i})$.

Note that while $y_v < 0$, it could be the case that $y'_v > 0$. If $y'_v > 0$, then we choose the point z on the line segment yy' which has $z_v = 0$. Evidently, this point can be written as a convex combination of $b^{\prec_i}$ $(i = 1, ..., k)$ and $b^{\prec_1^{u,e}}$ $(e \in (u, v]_{\prec_i})$. Otherwise, we choose $z = y'$. Before continuing on to the next iteration, we reduce the number of active bases in the convex combination for z to at most n as in Carathéodory's Theorem.

Theorem 4.2.4. *Given a submodular function g defined on S, at most n active orderings $\prec_1, \prec_2, ... \prec_k$ and coefficients $\lambda^1, \lambda^2, ..., \lambda^k$, the auxiliary digraph can be constructed, an arc uv can be selected for augmentation, and an augmentation can be performed in $O(n^3 + n^2q)$ operations, where $O(q)$ is the running time of the value-giving oracle for g.*

Proof. To construct the auxiliary digraph, for each active ordering $\prec$, and each pair $u, v \in S$, we check if $u \prec v$. There are $O(n)$ orderings and $O(n^2)$ pairs of elements for a total of $O(n^3)$ operations.

An arc can be chosen as follows. Add a supersource s to the graph and arcs se for each $e \in P$. Run breadth-first seach with source node s. This takes $O(n^2)$ time. To choose v, select from among the elements of N that element whose distance from s is greatest, and subject to this, choose the greatest element. This takes $O(n)$ time. Similarly, the greatest element among the elements $u \in S$ with uv an arc and $d(P, u) = d(P, v) - 1$ can be selected in $O(n)$ operations.

Subroutine 6, as described, requires $O(n^2q)$ operations. After choosing an arc uv and ordering $\prec$ for the augmentation step, the algorithm must compute the vectors $b^{\prec^{u,e}} - b^{\prec}$ for each $e \in (u, v]_{\prec}$. There are $O(n)$ such vectors, and $O(n)$ entries in each vector, each of which can be computed by querying the value-giving oracle.

Finding a point between y and y' requires $O(n)$ operations, and reducing the number of terms in the convex combination to at most n requires $O(n^3)$ operations. □

It remains to show that the algorithm performs polynomially many iterations. Let D be the auxiliary digraph in iteration i, and D' be the auxiliary digraph in iteration $i + 1$. Suppose that, in iteration i, the algorithm chooses the arc uv and ordering $\prec_1$ on which to perform an augmentation. Let α be defined as above and let

$$\beta = \text{number of } i \in \{1, ..., k\} \text{ with } |(u, v]_{\prec_i}| = \alpha.$$

Furthermore, suppose that, in iteration $i+1$, the algorithm chooses the arc $u'v'$. Let α' and β' be the quantities α and β in iteration $i+1$, and let P' be the set P in iteration $i+1$. We consider the tuples $(d(P,v), v, u, \alpha, \beta)$ and $(d'(P',v'), v', u', \alpha', \beta')$. We will need necessary preconditions for an appearance of an arc after an augmentation.

Lemma 4.2.5. *If pq is an arc of D' but not of D, then $u \preceq_1 q \prec_1 p \preceq_1 v$.*

Proof. Since pq is not an arc of D, we have $q \prec_1 p$. Since pq is an arc of D', we have $p \prec_1^{u,x} q$ for some $x \in (u,v]_{\prec_1}$. It follows that $p, q \in [u,v]_{\prec_1}$, as required. $\square$

We are now prepared to prove a monotonicity lemma.

Lemma 4.2.6. *For all $q \in S$, $d'(P',q) \geq d(P,q)$.*

Proof. Suppose not, and let $q \in S$ be such that $d'(P',q) < d(P,q)$. Subject to this, choose q so that $d'(P',q)$ is as small as possible. Note that, by the design of the algorithm, $P' \subseteq P$. Consequently, there is an arc pq of D' but not D such that $d(P,p) \leq d(P,q) - 2$. By Lemma 4.2.5, $u \preceq_1 q \prec_1 p \preceq_1 v$. It follows that $d(P,p) \leq d(P,q)+1$. But now,

$$d(P,q) \leq d(P,u) + 1 = d(P,v) \leq d(P,p) + 1 \leq d'(P',q),$$

a contradiction. $\square$

Lemma 4.2.7. *If $d'(P',e) = d(P,e)$ for all $e \in S$, then $(d(P,v), v, u, \alpha, \beta)$ is lexicographically greater than $(d'(P',v'), v', u', \alpha', \beta')$.*

Proof. Suppose $d(P,e) = d'(P',e)$ for each $e \in S$. Since $v' \in N'$, either $v' \in N$ or $v' = u$. Since $d(P,u) < d(P,v)$, $d'(P',v') \leq d(P,v)$. Moreover, if $d'(P',v') = d(P,v)$, then $v' \leq v$.

Assume $d'(P',v') = d(P,v)$ and $v' = v$. Then (u',v) is an arc of D'. If (u',v) is not an arc of D, then by Lemma 4.2.5, we have $u \preceq_1 v \prec_1 u' \preceq_1 v$, a contradiction. It follows from the choice of u that $u' \leq u$.

Now assume that $d'(P',v') = d(P,v)$, $v' = v$ and $u' = u$. Note that, for each $e \in (u,v]_{\prec_1}$, the set $(u,v]_{\prec_1^{u,e}}$ is a proper subset of $(u,v]_{\prec_1}$. Therefore, the augmentation step replaces an active base with one or more active bases, and for each such base, the size of the interval $(u,v]$ does not increase. It follows that $\alpha' \leq \alpha$.

Finally, assume $\alpha' = \alpha$. Note that $v \in N'$ and consider the augmentation step. The penultimate operation in the augmentation step is to choose a point z on the line segment between the old element of B_g (previously referred to as y) and the new element of B_g (previously referred to as y'). Recall that z is chosen as the closest point to y' with $z_v \leq 0$. Since $v \in N'$, it follows that $y'_v < 0$ and hence the algorithm must have chosen the point $z = y'$. Observe that the ordering $\prec_1$ does not occur in the convex combination for y'. Hence, the ordering $\prec_1$ is inactive in iteration $i+1$. Since any newly active ordering $\prec$ satisfies $|(u,v]_{\prec}| < |(u,v]_{\prec_1}|$, it follows that $\beta' < \beta$. □

All that remains is to determine is the running time of the algorithm.

Theorem 4.2.8. *Given a submodular function g defined on a finite set S, the pair (y, A) returned by Algorithm 7 is optimal in the Submodular Min-Max Theorem 4.1.1. Moreover, Algorithm 7 has a running time of $O(n^9 + n^8 q)$.*

Proof. By Theorem 4.2.2, the pair (y, A) is optimal in the Submodular Min-Max Theorem 4.1.1.

For the running time, first consider the vector $(d(P,e) : e \in S) \in \mathbb{Z}^S$. Since $0 \leq d(P,e) \leq n$ (where $d(P,e) = n$ if no path exists) for all $e \in S$, this vector assumes $O(n^2)$ values during the course of the algorithm. By Lemma 4.2.6, this vector is nondecreasing as the algorithm progresses. It follows that the set of all vectors in $\mathbb{Z}^S$ between $[0,0,...,0]^\top$ and $[n,n,...,n]^\top$ induce a partition of the iterations of the algorithm. Furthermore, each part consists of a series of consecutive iterations, which we will refer to as a *stage*.

Note that $u, v \leq n$ and since the augmentation step finishes by reducing the number of active bases to at most n, we also have $\alpha, \beta \leq n$. It follows from Lemma 4.2.7 that there are $O(n^4)$ iterations in each stage, giving a total of $O(n^6)$ iterations. By Theorem 4.2.4, each iteration requires $O(n^3 + n^2 q)$ operations, for a total of $O(n^9 + n^8 q)$ operations, as required. □

4.3 Matroid Polyhedron Membership: Local Change Algorithm

In this section, we return to the matroid polyhedron membership problem. We start by studying how Schrijver's augmentation step works in this special case, and conclude by refining the algorithm to obtain a different algorithm for matroid polyhedron membership. Schrijver gives a similar algorithm in his text *Combinatorial Optimization* [22]. The algorithm is similar to Cunningham's algorithm in that it works by an augmenting path scheme. The main difference is that Schrijver's algorithm works with respect to a single arc of an augmenting path, and this simplifies the analysis considerably. The difference between the algorithm given here and that of Schrijver is that Schrijver's augmentation step is simpler, leading to a slower running time.

Recall that an instance consists of a matroid $M = (S, \mathcal{I})$ and a non-negative vector $x \in \mathbb{R}^S$, and we are asked to find a representation for x as a convex combination of independent sets of M or a set $A \subseteq S$ such that $x(A) > r(A)$. In the current context, we are working over the base polyhedron of M, and so any point y considered by the algorithm satisfies $y(S) = r(S)$. It might be the case that $x(S) < r(S)$, and in this case, the algorithm will compute a vector $y \in B_M$ such that $x \leq y$, providing $x \in P_M$.

Note that, by Lemma 1.3.2, the rank function r of M is submodular. Evidently, the component sum function $x(A) = \sum_{e \in A} x_e$ is submodular, as is $-x(A)$. It follows that $g(A) = r(A) - x(A)$ is a submodular function. If $x \in P_M$, then $x(A) \leq r(A)$ for all $A \subseteq S$. Therefore, $x \in P_M$ if and only if the minimizing set A satisfies $g(A) \geq 0$.

One part of Schrijver's algorithm which simplifies considerably in the present context is the notion of a base. Given an ordering $\prec$ on the elements of S, consider the vector $b^{\prec}$ defined by

$$\begin{aligned} b^{\prec}_e &= g(e_{\prec} + e) - g(e_{\prec}) \\ &= r(e_{\prec} + e) - r(e_{\prec}) - x_e \\ &= \begin{cases} 0 - x_e & : e \in cl(e_{\prec}) \\ 1 - x_e & : \text{otherwise.} \end{cases} \end{aligned}$$

Consider the augmentation step. Recall that the vectors $b^{\prec^{u,e}} - b^{\prec}$ for each $e \in (u, v]_{\prec}$

have the form (on the interval $[u,v]_\prec$)

$$\begin{array}{c} \\ u \\ e_1 \\ e_2 \\ e_3 \\ \cdots \\ e_k \\ v \end{array}
\begin{array}{c} \prec^{u,e_1} \\ \begin{bmatrix} - \\ + \\ 0 \\ 0 \\ \cdots \\ 0 \\ 0 \end{bmatrix} \end{array},
\begin{array}{c} \prec^{u,e_2} \\ \begin{bmatrix} - \\ - \\ + \\ 0 \\ \cdots \\ 0 \\ 0 \end{bmatrix} \end{array},
\begin{array}{c} \prec^{u,e_3} \\ \begin{bmatrix} - \\ - \\ - \\ + \\ \cdots \\ 0 \\ 0 \end{bmatrix} \end{array}, \ldots,
\begin{array}{c} \prec^{u,v} \\ \begin{bmatrix} - \\ - \\ - \\ - \\ \cdots \\ - \\ + \end{bmatrix} \end{array}.$$

Since the rank function is integer valued, these vectors are also. In the $\delta > 0$ case, all $+$ entries are equal to 1, and precisely one of the $-$ entries is equal to -1. It follows in particular that $u \in C(B^\prec, v)$, where $B^\prec$ is the basis of M generated by the order $\prec$.

We are now prepared to give a refinement of Schrijver's algorithm. As in Section 2.6, the algorithm works over the basis polyhedron of M, that is, the polyhedron B_M described by $\{y \in \mathbb{R}^S : y \geq 0,\ y(A) \leq r(A)\ \forall A \subseteq S,\ y(S) = r(S)\}$. Consequently, every active independent set will be a basis of S. In contrast to the algorithms of Chapter 2, the refinement works by making small 'local' changes, that is, an augmentation raises y_e and lowers y_f for some arc ef of the auxiliary digraph, as opposed to making changes with respect to a source-sink path.

Given a convex combination λ of bases of M, let $y = y(\lambda)$ be defined as before. Let $N = \{e \in S : y_e < x_e\}$ and $P = \{e \in S : y_e > x_e\}$. The auxiliary digraph $D(M, x, \lambda)$ has vertex set S and its arc set contains the arcs

$$ef \text{ for all pairs } e, f \in S \text{ such that } f \in C(B, e) \text{ for some active basis } B.$$

Let $u(e,f)$, $D(e,f)$, and $L_J(e,f)$ be defined as before, and let $\prec$ be an order on $D(e,f)$. The modified augmentation step is shown as Subroutine 8. The subroutine requires the vector λ and elements $e, f \in S$ with $e \in N$ as input.

Note that Subroutine 8 chooses δ as large as possible so that $y' \in P_M$ and $y' \geq y$. Note also that many independent sets can be involved in an augmentation. This is in contrast to Schrijver's algorithm in [22] which only performs augmentations with respect to a single

Subroutine 8 Local Augmentation of λ using ef

```
$\lambda' \leftarrow \lambda$
$\delta \leftarrow \min\{x_e - y_e, u(e,f)\}$
**for** each $J \in D(e,f)$ **do**
    $J' = J + e - f$
    **if** $L_J(e,f) \leq \delta$ **then**
        $\lambda'_J = 0$
        $\lambda'_{J'} = \lambda_{J'} + \lambda_J$
    **else if** $\delta < L_J(e,f) < \delta + \lambda_J$ **then**
        $\lambda'_J = L_J(e,f) - \delta$
        $\lambda'_{J'} = \lambda_{J'} + \delta - L_J(e,f) + \lambda_J$
    **end if**
**end for**
**return** $\lambda'$
```

independent set (and its corresponding mutation).

Trivially, Subroutine 8 produces a convex combination of bases of M, given such a combination. More importantly, one of two things has happened. In the $\delta = x_e - y_e$ case, the element e now satisfies $x_e = y_e$. In the $\delta = u(e,f)$ case, we have $x_e \leq y_e$ and ef is not an arc of the next auxiliary digraph. Observe that in both cases, the augmentation chooses δ so that no new elements are added to P.

Let D be the auxiliary digraph. Apply Subroutine 8 to the first arc ef of an (N,P)-dipath in D, and let D' be the next auxiliary digraph. We can prove the necessary preconditions for the appearance of a new arc in the auxiliary digraph directly from the Strong Circuit Axiom.

Proposition 4.3.1. *If ab is an arc of D' but not of D, then ($a = f$ or) af is an arc of D and ($b = e$ or) eb is an arc of D.*

Proof. Since ab is an arc of D' but not D, there exists $J \in D(e,f)$ such that $b \in C(J + e - f, a) \setminus C(J,a)$. By the Strong Circuit Axiom 1.3.4, there exists a circuit C_1 such that

$$\begin{aligned} b \in C_1 &\subseteq (C(J+e-f,a) \cup C(J,a)) - a \\ &\subseteq J + e. \end{aligned}$$

Hence, $b \in C_1 = C(J, e)$.

If $f \notin C(J, a)$, then again by the Strong Circuit Axiom 1.3.4, there exists a circuit C_2 such that

$$\begin{aligned} f \in C_2 &\subseteq (C(J, a) \cup C(J, e)) - b \\ &= J - b \cup \{a, e\} . \end{aligned}$$

Since $b \notin C(J, a)$, $f \in C_2 = C(J, a)$, a contradiction. □

Let d and d' be the distance functions of D and D', respectively. Let P' and N' be the sets P and N in D'. Note that $P' \subseteq P$. A monotonicity property now follows easily.

Lemma 4.3.2. *For all $u \in S$, $d'(u, P') \geq d(u, P)$.*

Proof. Assume $d'(u, P') < d(u, P)$, and choose u so that $d'(u, P')$ is minimized. Then there must be an arc uv of D' but not D with $d(v, P) \leq d'(v, P')$. By Proposition 4.3.1, $d(e, P) \leq d(v, P) + 1$ and $d(u, P) \leq d(f, P) + 1$. Now,

$$d(e, P) \leq d(v, P) + 1 \leq d'(v, P') + 1 = d'(u, P') < d(u, P) \leq d(f, P) + 1 = d(e, P),$$

a contradiction. □

Algorithm 9 Matroid Polyhedron Membership - Local Augmentation Version

```
$\lambda \leftarrow 0$
$B \leftarrow$ arbitrary basis of $S$
$\lambda_B \leftarrow 1$
$D \leftarrow D(M, x, \lambda)$
while there exists an $(N, P)$-dipath in $D$ do
    $e \leftarrow$ greatest element of $N$ with $d(e, P)$ maximum
    $f \leftarrow$ greatest element of $S$ with $d(f, P) = d(e, P) - 1$ and $f \in C(B, e)$ with $\lambda_B > 0$
    $\lambda \leftarrow$ result of Subroutine 8 using $e, f$ and $\lambda$
    reduce number of active bases to at most $n$
    $D \leftarrow D(M, x, \lambda)$
end while
$A \leftarrow \{e \in S :$ there exists an $(N, e) -$ dipath in $D\}$
return $(\lambda, A)$
```

The local version of the algorithm is given as Algorithm 9. The algorithm requires a matroid $M = (S, \mathcal{I})$, a vector $x \in \mathbb{R}^S$, and any total order on S as inputs. By the greatest element of S, or the greatest element of N, we mean the greatest element according to the given total order.

Lemma 4.3.3. *If $d'(e, P') = d(e, P)$ for all $e \in S$, then $e \geq e'$, $f \geq f'$, and if $e = e'$, then $f > f'$.*

Proof. Assume that $d'(e, P') = d(e, P)$ for all $e \in S$. Then $P' = P$. By the choice of e, $d'(e', P') = d(e', P) \leq d(e, P)$. Assume equality holds. By the choice of e, $e' \leq e$. Assume equality holds again. Then the augmentation step chose amount $\delta = u(e, f) < x_e - y_e$, so ef is not an arc of D'. If $f' > f$, then ef' is an arc of D' but not D. By Proposition 4.3.1, it must be the case that $f' = e$, a contradiction. $\square$

For a vector λ returned by the algorithm, let $y = y(\lambda)$ be defined as before, and let y' be defined by $y'_e = \min\{y_e, x_e\}$.

Theorem 4.3.4. *Given a matroid $M = (S, \mathcal{I})$ and a vector $x \in \mathbb{R}^S$ and any total order on S, Algorithm 9 returns (λ, A) such that (y', A) is an optimal pair in the Membership Min-Max Theorem 2.1.2. Moreover, Algorithm 9 is a polynomial time matroid algorithm with a running time of $O(n^7 \cdot q)$.*

Proof. Let (λ, A) be the pair returned by Algorithm 9. Since $N \subseteq A$ and $P \cap A = \emptyset$, $y'(S \setminus A) = x(S \setminus A)$ and $y'(A) = y(A)$. We claim that, for each active basis B, $A \cap B$ is a basis for A. If not, then there exists some $e \in A \setminus B$ such that $(A \cap B) + e \in \mathcal{I}$. Since B is a basis, it follows that there exists some $f \in B \setminus A$ such that $f \in C(B, e)$, a contradiction. Now,

$$y(A) = \sum_{e \in A} y_e = \sum_{B \in \mathcal{I}} \lambda_B \cdot |A \cap B| = \sum_{B \in \mathcal{I}} \lambda_B \cdot r(A) = r(A).$$

So $y'(A) = y(A) = r(A)$ and $y'(S \setminus A) = x(S \setminus A)$, as required.

For the running time, divide the iterations into stages, where each stage consists of all iterations where $d(e, P)$ is constant for all $e \in S$. Then there are $O(n^2)$ stages. By Lemma 4.3.3, there are $O(n^2)$ iterations per stage, and hence $O(n^4)$ iterations in total. The time to perform an iteration is dominated by the time to build the auxiliary digraph, which is $O(n^3 \cdot q)$, giving the bound. $\square$

4.4 Matroid Polyhedron Membership: Push-Relabel Algorithm

In this section, we study a push-relabel algorithm for the matroid polyhedron membership problem, due to Frank and Miklós [12]. The algorithm is apparently very similar to the refinement of Schrijver's [21] algorithm presented in the previous section, though it has a simpler selection rule and does not use an auxiliary digraph. The push-relabel algorithm is a somewhat natural extension of the algorithm of the previous section. Notice how that algorithm constructs the auxiliary digraph once for each iteration, but uses only very little information to make its selection of elements. In fact, it only needs some $e \in N$ and some $f \in S$ with $d(f, P) = d(e, P) - 1$. The push-relabel algorithm capitalizes on this observation by maintaining a labelling of the elements of S which approximates the distance function of the auxiliary digraph.

Let $M = (S, \mathcal{I})$ and $x \in \mathbb{R}^S$ be an instance of the matroid polyhedron membership problem. As before, the algorithm keeps a convex combination $\lambda \in \mathbb{R}^{\mathcal{I}}$ of bases of M. Given such a vector λ, let $y = y(\lambda)$ be defined as before. Define the sets $P = \{e \in S : \ y_e > x_e\}$ and $N = \{e \in S : \ y_e < x_e\}$.

In addition, the algorithm keeps a level function $\mathcal{L} : S \to \{0, 1, 2, ..., n\}$. To gain an intuitive understanding of the level function, and indeed the algorithm itself, it is helpful to consider the algorithm of the previous section and also Cunningham's algorithm for matroid polyhedron membership. The push-relabel algorithm, like the algorithm of the previous section, works by making local changes in the form of basis exchanges. The augmentation step itself consists of repeated single basis exchanges, and is easily seen to be equivalent to the augmentation step of the previous section's algorithm. As an advantage, it does not require an ordering on the elements of S for its selection rule. When e and f are exchanged, the values of y_e and y_f are, respectively, increased and decreased by some $\delta > 0$.

In the basis version of Cunningham's algorithm, the algorithm seeks a sequence $e_1, e_2, ..., e_k$ of elements of S such that $e_1 \in N$, $e_k \in P$ and such that, for each $i \in \{1, ..., k-1\}$, there exists a basis B_i with $\lambda_{B_i} > 0$ and $e_{i+1} \in C(B_i, e_i)$. The algorithm then applies the augmentation step, which increases y_{e_1} and decreases y_{e_k}. The idea of the level function is that it places a lower bound on the number of elements in such a sequence. By managing this level function, and by a consistent choice of elements for augmentation, it is possible to simplify the numerous rules for choosing elements in the local version of the algorithm.

Algorithm 10 Push-Relabel Matroid Polyhedron Membership Algorithm

```
$\lambda \leftarrow 0$
$B \leftarrow$ arbitrary basis of $S$
$\lambda_B \leftarrow 1$
$\mathcal{L}(a) \leftarrow 0$ for all $a \in S$
while there exists $a \in N$ with $\mathcal{L}(a) \leq n-1$ do
    $e \leftarrow$ any element of $N$ for which $\mathcal{L}(e) \leq n-1$ is maximum
    while $x_e < y_e$ and a push at $e$ is possible do
        $\lambda \leftarrow$ result of push at $e$
    end while
    if $x_e = y_e$ then
        relabel $e$
    end if
end while
Choose $i$ so that $\{e \in S : \mathcal{L}(e) = i\} = \emptyset$
$A \leftarrow \{e \in S : \mathcal{L}(e) > i\}$
return $(\lambda, A)$
```

The push-relabel algorithm, due to Frank and Miklós [12], is given as Algorithm 10. The algorithm relies on two simple operations, push and relabel. To *relabel* e is to increase $\mathcal{L}(e)$ by 1. For some $e \in N$, a push at e is possible whenever there exists some $f \in S$ such that $\mathcal{L}(f) = \mathcal{L}(e) - 1$ and there exists an active basis B such that $f \in C(B, e)$. The *push* operation chooses such an element f, basis B, and the quantity $\delta = \min\{y_e - x_e, \lambda_B\}$. It then reduces and increases (respectively) the coefficients of B and $B + e - f$ by δ. A push at e which sets $y_e = x_e$ is called a *neutralizing push*.

Proposition 4.4.1. *Throughout the course of execution, the level function maintains the following invariants.*

(i) $\mathcal{L}(e) = 0$ *for all* $e \in P$.

(ii) *For every active basis* B *and every* $e \in S \setminus B$, $\min_{f \in C(B,e)} \{\mathcal{L}(f)\} \geq \mathcal{L}(e) - 1$.

Proof. First observe that the push operation chooses δ so that $e \notin P$ after a push. Hence, no element of S is ever added to P. Moreover, the algorithm chooses only members of N for the push operation, and so only relabels members of N. It follows that $\mathcal{L}(e) = 0$ for each $e \in P$.

Evidently, (ii) holds at initialization. Consider the earliest point during execution where (ii) is violated by the active basis B, the element $e \in S \setminus B$ and the element $f \in C(B, e)$ satisfying $\mathcal{L}(f) < \mathcal{L}(e) - 1$. Since (ii) was satisfied before the previous operation, either e was relabelled or the coefficient of B was increased from 0. The former case is not possible, since the algorithm would not have relabelled e if the coefficient of B was positive. Thus, we may assume that the previous operation was a push.

In the latter case, $B = B' + a - b$, where the last push was performed at a with respect to b. Moreover, $f \in C(B, e) \setminus C(B', e)$. By the Strong Circuit Axiom (1.3.4), there exists a circuit C of M such that

$$f \in C \subseteq (C(B' + b - a, e) \cup C(B', e)) - e \subseteq B' + a.$$

It follows that $f \in C = C(B', a)$. Since $\mathcal{L}(e) - 1 > \mathcal{L}(f)$ and since (ii) was satisfied by B', it must be the case that $\mathcal{L}(e) - 1 > \mathcal{L}(f) \geq \mathcal{L}(a) - 1$. By the choice of b, $\mathcal{L}(b) = \mathcal{L}(a) - 1$.

Consider the circuit $C(B', e) = C(B - a + b, e) = C(B - a + e, b)$; it follows that $b \in C(B', e)$. Since B' satisfies (ii), it must also be that $\mathcal{L}(b) \geq \mathcal{L}(e) - 1$. But now,

$$\mathcal{L}(b) \geq \mathcal{L}(e) - 1 > \mathcal{L}(f) \geq \mathcal{L}(a) - 1 = \mathcal{L}(b),$$

a contradiction. It follows that the property (ii) is invariant. □

If $N = \emptyset$ after an iteration, then $y \in B_M$ and $y \geq x$, so $x \in P_M$. Otherwise, the algorithm seeks to transfer the deficit (that is, $x_e - y_e$) at some $e \in N$ to some collection of members of P. It does this by moving some amount $\delta \leq x_e - y_e$ of the deficit at e to an element whose level is $\mathcal{L}(e) - 1$. In some subsequent iteration, the algorithm will then move some or all of δ to an element at a lower level, and so on. Assuming $x \in P_M$, the deficit eventually makes its way down to level 1. The algorithm then uses the surplus (that is, $y_f - x_f$) at the elements $f \in P$ to neutralize the deficit.

This brings us to the case where $x \notin P_M$. In this case, the algorithm should reach a state where it is provably impossible to transfer the deficit from any $e \in N$ to any $f \in P$. Intuitively, this happens when there is a level which 'separates' N from P. That is, a level i which is empty, and which is below the level of any member of N. Since the algorithm only relabels elements of N, no element will enter level i, and hence no deficit can be transferred from a member of N to any other element whose level is less than $i+1$. Formally, the termination condition is

there exists $i \in \{0, 1, ..., n\}$ *such that* $\{e \in S : \mathcal{L}(e) = i\} = \emptyset$ *and for all* $f \in N$, $\mathcal{L}(f) > i$.

Lemma 4.4.2. *If the termination condition holds, then the vector* y' *defined by* $y'_e = \min\{y_e, x_e\}$ *and the set* $A = \{e \in S : \mathcal{L}(e) > i\}$ *are an optimal pair in the Membership Min-Max Theorem 2.1.2.*

Proof. It suffices to show that $y'(S) = r(A) + x(S \setminus A)$. For each $e \in S \setminus A$, $e \notin N$. It follows that $y'_e = x_e$, and hence $y'(S \setminus A) = x(S \setminus A)$. Note that $y'(A) = y(A)$. By property (ii) of the level function $\mathcal{L}$, for each active basis B and each $e \in A \setminus B$, the circuit $C(B, e)$ is contained in A. It follows that $A \cap B$ is a basis of A for each active basis B. Now,

$$\begin{aligned} y(A) &= \sum_{B \in \mathcal{B}} \lambda_B \cdot |B \cap A| \\ &= \sum_{B \in \mathcal{B}} \lambda_B \cdot r(A) \\ &= r(A). \end{aligned}$$

Thus, $y'(S) = y(A) + y'(S \setminus A) = r(A) + x(S \setminus A)$, as required. □

If $y \ngeq x$, then $N \neq \emptyset$. If every $e \in N$ satisfies $\mathcal{L}(e) = n$, then by the Pigeonhole Principle, the termination condition holds. Otherwise, the algorithm is able to select some $e \in N$ with $\mathcal{L}(e) < n$ and proceed with an iteration. On the other hand, if $y \geq x$, then $N = \emptyset$, so the termination condition holds, either for level n or for some other level.

Theorem 4.4.3. *Given a matroid* $M = (S, \mathcal{I})$ *and a vector* $x \in \mathbb{R}^S$, *Algorithm 10 returns* λ *and* A *such that* (y', A) *is an optimal pair in the Membership Min-Max Theorem 2.1.2. Moreover, Algorithm 10 is a polynomial-time matroid algorithm with a running time of* $O(n^7 \cdot q)$.

Proof. The pair (y', A) is optimal in the Membership Min-Max Theorem by Lemma 4.4.2.

For the running time, divide the iterations into stages, where each stage consists of all iterations where $\mathcal{L}(e)$ is constant for all $e \in S$. Evidently, there are $O(n^2)$ stages. Since the algorithm chooses an element $e \in N$ to maximize $\mathcal{L}(e)$, e is neutralized at most once per stage. Hence, there are $O(n^3)$ neutralizing pushes in all. In a non-neutralizing push, λ_B is set to 0. Thus, the number of active bases does not increase. Similarly, in a neutralizing push, the number of active bases increases by at most one. It follows that the number of active bases is $O(n^3)$. Before performing a neutralizing push at e, it may be necessary to perform a push at e for each active basis. As there are $O(n^3)$ such bases, there are $O(n^6)$ pushes in all.

Given a basis B, an element e and the set $L = \{f \in S : \mathcal{L}(f) = \mathcal{L}(e) - 1\}$, $O(n)$ queries of the independence oracle are required to identify a candidate $f \in C(B, e) \cap L$. The basis exchange itself can be performed in constant time, giving the algorithm an overall running time of $O(n^7 \cdot q)$, where $O(q)$ is the running time of the independence oracle. □

As before, the running time of the algorithm can be improved by reducing the number of active bases to at most n at the end of each stage, per Carathéodory's Theorem. A total of $O(n^2)$ such reductions will be performed, each in $O(n^3)$ operations. The number of active bases is then reduced to $O(n)$, which means there are $O(n^4)$ pushes in all, giving a running time of $O(n^5 \cdot q)$.

4.5 Push-Relabel Submodular Function Minimization

The push-relabel algorithm of Frank and Miklós [12] presented in the previous section is very similar to the matroid polyhedron membership version of Schrijver's algorithm presented in section 4.3. Indeed, the level function $\mathcal{L}$ serves only to simplify the rules for choosing a pair (u, v) for augmentation. Frank and Miklós extend their algorithm to solve the polymatroid intersection problem, of which submodular minimization is a special case. In this section, we consider the algorithm obtained by applying the ideas of Frank and Miklós to Schrijver's algorithm.

Although the presentation is different, the algorithm turns out to be identical to an algorithm given by Fleischer and Iwata in 2003 [10]. The algorithm keeps a level function $\mathcal{L} : S \to \{0, 1, 2, ..., n\}$ and begins by setting $\mathcal{L}(e) = 0$ for all $e \in S$, choosing a random ordering $\prec$ on S, and setting the coefficient of $\prec$ to 1. At each iteration, the algorithm chooses some element $v \in S$ with $\mathcal{L}(v) \leq n - 1$ maximum. If there exists an element $u \in S$ with $\mathcal{L}(u) = \mathcal{L}(v) - 1$ and such that $u \prec v$ for some active ordering $\prec$, then a push at v is possible. A push at v with respect to u, the ordering $\prec$, and the point $y \in B_g$

(represented as a convex combination of bases) consists of applying Subroutine 6 to u, v and $\prec$ to obtain a point $y' = y + \delta(\chi_v - \chi_u)$, then choosing the point z on the line segment yy' maximizing z_v such that $z_v \leq 0$, and updating the convex combination of bases so that it represents z. The last step of the push consists of reducing the number of active bases in the convex combination to at most n. The push operation is defined in Subroutine 11 and the push-relabel algorithm is defined in Algorithm 12.

Subroutine 11 Push at v with respect to u, $\prec_i$, and y

```
$\kappa \leftarrow$ result of Subroutine 6 with $u, v$ and $\prec_i$
$y' \leftarrow \sum_{j \neq i} \lambda^j b^{\prec_j} + \lambda^i \sum_{e \in (u,v]_{\prec_i}} \kappa_e b^{\prec_i^{u,e}}$
if $y'_v > 0$ then
    $y \leftarrow$ point $z$ on line segment $yy'$ with $z_v = 0$
else
    $y \leftarrow y'$
end if
reduce number of active bases in convex combination for $y$
return convex combination for $y$
```

Proposition 4.5.1. *During the course of execution, the level function maintains the following invariants.*

(i) $\mathcal{L}(u) = 0$ *for all* $u \in P$.

(ii) For all active orderings $\prec$ *and all* $u, v \in S$ *with* $u \prec v$, $\mathcal{L}(u) \geq \mathcal{L}(v) - 1$.

Proof. Evidently, the property (i) holds at initialization. Since only elements of N are relabelled, and since the augmentation step never adds a new element to P, it follows that (i) is invariant.

Again, (ii) holds at initialization. Consider the first iteration where (ii) is violated for some active ordering $\prec$ and $u, v \in S$ with $u \prec v$. Since (ii) held in the previous iteration, it held with respect to u, v and every active ordering. Note that the algorithm could not have relabelled u or v at the end of the previous iteration, so it must be the case that $\prec$ was not active in the previous iteration. Suppose e and f were chosen for augmentation in the previous iteration. Then by Lemma 4.2.5, there was an active ordering $\prec_1$ such that $e \preceq_1 v \prec_1 u \preceq_1 f$. Since (ii) was satisfied, we have

$$\mathcal{L}(e) \geq \mathcal{L}(v) - 1 > \mathcal{L}(u) \geq \mathcal{L}(f) - 1 = \mathcal{L}(e),$$

Algorithm 12 Push-Relabel Submodular Function Minimization Algorithm

```
$\prec_1 \leftarrow$ arbitrary order on $S$
$\lambda^1 \leftarrow 1$
$y \leftarrow b^{\prec_1}$
$\mathcal{L}(a) \leftarrow 0$ for all $a \in S$
while there exists $a \in N$ with $\mathcal{L}(a) \leq n-1$ do
    $v \leftarrow$ any element of $N$ for which $\mathcal{L}(v) \leq n-1$ is maximum
    while $v \in N$ and a push at $v$ is possible do
        $u \leftarrow$element of $S$ with $\mathcal{L}(u) = \mathcal{L}(v) - 1$ and $u \prec v$ for active $\prec$
        while $u \prec v$ for some active $\prec$ and $y_v < 0$ do
            $\prec \leftarrow$active order with $u \prec v$ maximizing $|(u, v]_{\prec}|$
            $y \leftarrow$ result of push at $v$ with respect to $u$ and $\prec$
        end while
    end while
    if $v \in N$ then
        increase $\mathcal{L}(v)$ by 1
    end if
end while
Choose $i$ such that $\{e \in S : \mathcal{L}(e) = i\} = \emptyset$
$A = \{e \in S : \mathcal{L}(e) > i\}$
return $(y, A)$
```

a contradiction. It follows that the property (ii) is invariant. □

The algorithm continues to run while there exists some $e \in N$ such that $\mathcal{L}(e) \leq n-1$. Therefore, at termination, every $e \in N$ has level n.

Theorem 4.5.2. *If every $e \in N$ satisfies $\mathcal{L}(e) = n$, then there exists a level i $(0 \leq i \leq n)$ such that $\{e \in S : \mathcal{L}(e) = i\} = \emptyset$ and such that for each $e \in N$, $\mathcal{L}(e) > i$. Moreover, the set*

$$A = \{e \in S : \mathcal{L}(e) > i\}$$

and the vector y are an optimal pair in the Submodular Min-Max Theorem (4.1.1).

Proof. The first assertion follows directly from the Pigeonhole Principle. For the second assertion, observe that $N \subseteq A$ and $P \cap A = \emptyset$, so for any $R \subseteq S$, we have $y(A) \leq y(R)$; note also that $y(A) = y^-(S)$. As in Theorem 4.2.2, for each active ordering $\prec_i$, $b^{\prec_i}(A) = g(A)$. Now let $R \subseteq S$. We have

$$\begin{aligned} y(A) &= y^-(S) \\ &= \sum_{e \in A} \sum_{i=1}^{k} \lambda^i b_e^{\prec_i} \\ &= \sum_{i=1}^{k} \lambda^i b^{\prec_i}(A) \\ &= g(A), \end{aligned}$$

as required. □

All that remains is to determine the running time of the algorithm. Recall that a push at v is called neutralizing if it sets $y_v = 0$; otherwise, it is non-neutralizing.

Lemma 4.5.3. *Algorithm 12 terminates after $O(n^6)$ pushes.*

Proof. Consider the innermost loop of Algorithm 12. Define

$$\alpha = \max_{\text{active } i} \{|(u,v]_{\prec_i}|\} \quad \text{and} \quad \beta = \text{number of active } i \text{ with } |(u,v]_{\prec_i}| = \alpha.$$

A non-neutralizing push decreases one of the above quantities. Moreover, the sequence of α and β values generated by repeated (and uninterrupted) iterations of the loop are lexicographically decreasing. It follows that the innermost loop terminates after $O(n^2)$ iterations.

Now consider the inner loop of Algorithm 12. By Lemma 4.2.5, each element $u \in S - v$ can be selected at most once without breaking the loop. Furthermore, this loop terminates with either a neutralizing push at v or by relabeling v.

By the choice of v (element of N with $\mathcal{L}(v)$ maximum), and the choice of u ($\mathcal{L}(u) = \mathcal{L}(v) - 1$), each element $v \in N$ can be neutralized at most once between two relabel operations. By the above discussion, there are $O(n^3)$ non-neutralizing pushes between two neutralizing pushes, and $O(n)$ neutralizing pushes between two relabels. Clearly, there are $O(n^2)$ relabels, giving $O(n^6)$ pushes in all. $\square$

Theorem 4.5.4. *Given a submodular function g defined on S, Algorithm 12 returns an optimal pair in the Submodular Min-Max Theorem 4.1.1. Moreover, the algorithm has running time $O(n^9 + n^8 \cdot q)$, where $O(q)$ is the running time of the value-giving oracle for g.*

Proof. By Theorem 4.5.2, the pair (y, A) returned by Algorithm 12 is optimal in Theorem 4.1.1. By Lemma 4.5.3, there are $O(n^6)$ calls to Subroutine 11, whose running time is $O(n^3 + n^2 \cdot q)$, giving the bound. $\square$

References

[1] R. E. Bixby, W. H. Cunningham, and D. M. Topkis, *The partial order of a polymatroid extreme point*, Mathematics of Operations Research **10** (1985), 367–378.

[2] J.A. Bondy and U. S. R. Murty, *Graph Theory*, Springer, New York, London, 2007.

[3] William H. Cunningham, *Testing membership in matroid polyhedra*, Journal of Combinatorial Theory, Series B **36** (1984), 161–188.

[4] ______, *On submodular function minimization*, Combinatorica **5** (1985), 185–192.

[5] ______, *Improved bounds for matroid partition and intersection algorithms*, SIAM Journal on Computing **15** (1986), 948–957.

[6] E. A. Dinic, *Algorithm for solution of a problem of maximum flow in a network with power estimation*, Soviet Math Doklady **11** (1970), 1277–1280.

[7] Jack Edmonds, *Submodular functions, matroids, and certain polyhedra*, in (R.K. Guy et al.) Combinatorial Structures and Their Applications (New York), Gordon and Breach, 1970, pp. 69–87.

[8] ______, *Matroids and the greedy algorithm*, Mathematical Programming **1** (1971), 127–136.

[9] Jack Edmonds and Richard M. Karp, *Theoretical improvements in algorithmic efficiency for network flow problems*, J. ACM **19** (1972), 248–264.

[10] Lisa Fleischer and Satoru Iwata, *A push-relabel framework for submodular function minimization and applications to parametric optimization*, Discrete Applied Mathematics **131** (2003), 311 – 322.

[11] L. R. Ford and D. R. Fulkerson, *Maximal flow through a network.*, Canadian Journal of Mathematics **8** (1956), 399–404.

[12] András Frank and Zoltán Miklós, *Simple push-relabel algorithms for matroids and submodular flows*, Japan Journal of Industrial and Applied Mathematics **29** (2012), 419–439.

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

[14] Julian Haselmayr, *Schnitt von matroiden theorie und algorithmen*, Diploma thesis, University of Augsberg, 2008.

[15] John E. Hopcroft and Richard M. Karp, *An $O(n^{\frac{5}{2}})$ algorithm for maximum matchings in bipartite graphs*, SIAM Journal on Computing **2** (1973), 225–231.

[16] Satoru Iwata, Lisa Fleischer, and Satoru Fujishige, *A combinatorial strongly polynomial algorithm for minimizing submodular functions*, J. ACM **48** (2001), 761–777.

[17] L.G. Khachiyan, *Polynomial algorithms in linear programming*, USSR Computational Mathematics and Mathematical Physics **20** (1980), 53 – 72.

[18] E. L. Lawler and C. U. Martel, *Computing maximal "polymatroidal" network flows.*, Mathematics of Operations Research **7** (1982), 334 – 347.

[19] James G. Oxley, *Matroid Theory*, second ed., Oxford Graduate Texts in Mathematics, Oxford University Press, 2011.

[20] Paul Schönsleben, *Ganzzahlige polymatroid-intersektions-algorithmen*, Ph.D. thesis, ETH Zürich, 1980.

[21] Alexander Schrijver, *A combinatorial algorithm minimizing submodular functions in strongly polynomial time*, Journal of Combinatorial Theory, Series B **80** (2000), 346 – 355.

[22] Alexander Schrijver, *Combinatorial Optimization - Polyhedra and Efficiency*, Springer, 2003.

Index