Report: Polyhedral Combinatorics, Matroids and Derandomization of Isolation Lemma

Instructor: Rohit Gurjar

Soham Chatterjee

sohamchatterjee999@gmail.com
Website: sohamch08.github.io

2024

Contents

Chapter 1

Introduction

1.1 Some Basics of Graph Theory

CHAPTER 2

Matorids

2.1 Matroids

Definition 2.1.1: Matroid

A matroid $M = (E, \mathcal{I})$ has a ground set E and a collection I of subsets of E called the *Independent Sets* st

1. Downward Closure: If $Y \in \mathcal{I}$ then $\forall\, X \subseteq Y$, $X \in \mathcal{I}$.
2. Extension Property: If $X, Y \in \mathcal{I}$, $|X| < |Y|$ then $\exists\, e \in Y - X$ such that $X \cup \{e\}$ also written as $X + e \in \mathcal{I}$

Observation. *A maximal independent set in a matroid is also a maximum independent set. All maximal independent sets have the same size.*

Base: Maximal Independent sets are called bases.
Rank of $S \in I$: We define the rank function of a matroid $r : \mathcal{P}(E) \to \mathbb{Z}$ where $r(S) = \max\{|X| \colon X \subseteq S, X \in I\}$ We def
Rank of a Matroid: Size of the base.
Span of $S \in I$: $\{e \in E \colon rank(S) = rank(S + e)\}$

2.2 Examples of Matroids

Uniform Matroid: It is denoted as $U_{k,n}$ where $E = [n]$ and $I = \{X \subseteq E \mid |X| \leq k\}$.
Free Matroid: When $k = n$ we take all possible subsets of E into I. This matroid is called Free Matroid i.e. $U_{n,n}$
Partition Matroid: Given $E = E_1 \sqcup E_2 \sqcup \cdots \sqcup E_l$ where $\{E_1, \ldots, E_l\}$ is a partition of E and $k_1, \ldots, k_l \in \mathbb{N} \cup \{0\}$

$$I = \{X \subseteq E \colon |X \cap E_i| \leq k_i\ \forall\, i \in [l]\}$$

then $M = (E, I)$ is a partition matroid.

Note:-

If the E_i's are not a partition then suppose E_1, E_2 has nonempty partition then we will not have a matroid.
For example: $E_1 = \{1, 2\}$, $E_2 = \{2, 3\}$ and $k_1 = k_2 = 1$ then $X = \{1, 3\}$ is independent but $Y = \{2\} \subsetneq X$ is not a matroid.

Linear Matroid: Given a $m \times n$ matrix denote its columns as $A_1, \ldots, A_n$. Then

$$I = \{X \subseteq [n] \colon \text{Columns corresponding to } X \text{ are linearly independent}\}$$

Here if the underlying field is $\mathbb{F}_2$ then it is called *Binary Matroid* and for $\mathbb{F}_3$ it is called *Ternary Matroid.*
Representable Matroid: A matroid with which we can associate a linear matroid is called a representable matroid.

Eg: $U_{2,3}$. It can be represented by the matrix $A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$, over $\mathbb{F}_2$. Over $\mathbb{F}_3$ it is same as $U_{3,3}$.

Note:-

There are matroids which are not representable as linear matroids in some field. There are matroids which are not representable on any field as well.

Lemma 2.2.1

$U_{2,4}$ is not representable over $\mathbb{F}_2$ but representable over $\mathbb{F}_3$

Regular Matroid: There are the matroids which are representable over all fields.

Lemma 2.2.2

Regular Matroids are precisely those which can be represented over $\mathbb{R}$ by a Totally Uni-modular matrix

Graphic Matroid / Cyclic Matroid: For a graph $G = (V, E)$ the graphic matroid $M_G = (E, I)$ where

$$I = \{F \subseteq E \colon F \text{ is acyclic}\}$$

Hence I is the collection of forests of G. It follows the downward closure trivially. For extension property let $k = |F_1| < |F_2| = l$ and then there are $n - k$ and $n - l$ components. So $n - k > n - l$. So $\exists$ an edge in F_2 which joins 2 components in F_1.

Lemma 2.2.3

A subset of columns is linearly independent iff the corresponding edges don't contain a cycle in the incidence matrix

Lemma 2.2.4

Graphic Matroids are Regular Matroids

Proof Idea: Use Incidence Matrix. ■

Matching Matroids: We can try to define it like this but it will not work:

Problem 2.1

Is the following a matroid: $E =$ Edges of a graph and $I = \{F \subseteq E \colon F \text{ is a matching}\}$

Solution: It is not a matroid since maximal matchings can not be extended to a maximum matching.

■

Correct way will be: For a graph $G = (V, E)$ the ground set $= V$ and

$$I = \{S \subseteq V \colon \exists \text{a matching that matches all vertices in } S\}$$

The downward closure property trivially holds. For extension property is $|S| < |S'|$ then there exists another vertex in S' which is not matched with S, so we can add that vertex to S.

2.3 Circuits

Assume we have a matroid $M = (E, I)$.

Definition 2.3.1: Circuit

A minimal dependent set C such that $\forall\, e \in C$, $C - e$ is an independent set.

Theorem 2.3.1

Let $S \in I$. $S + e \notin I$. Then $\exists!\, C \subseteq S + e$.

Proof: Given $S + e \notin I$. Take the set Σ where $T \in \Sigma$ if $t \notin I$ and $T \subseteq S + e$. Σ is nonempty since $S + e \in \Sigma$. Now under the ordering of inclusion T has a minimal element. Hence this minimal element is the desired circuit C which is minimal dependent set contained in $S + e$.

Now suppose it is not unique. Let $C_1, C_2 \subseteq S + e$ be circuits. Suppose $f \in C_1 - C_2$. Then $S - e + f$ will still be dependent since $C_2 \subseteq S - e + f$. Now by definition we get that $C_1 - f$ is independent. Therefore we extend $C_1 - f$ to an independent set by adding the elements of S till we reach same size as $|S|$. Now $e \in C$ since C_1 was formed because of addition of e. Hence if we extend $C_1 - f$ till same cardinality as S we will add all the edges of S not in $C_1 - f$ except f since adding f will make C be a dependent subset of an independent set which is not possible. Hence $C_1 - f$ will be extended to $S - f + e$. Therefore $S + e - f$ is independent which contradicts our previous conclusion that $S + e - f$ is dependent. Hence contradiction. ■

2.4 Axiom Systems for a Matroid

2.5 Finding Max Weight Base

The problem is given a matroid $M = (E, I)$ and a weight function $W : E \to \mathbb{R}$ find the maximum weight base of the matroid. We will solve this using basic greedy algorithm.

2.5.1 Algorithm

Algorithm 1: Algorithm for Finding Max Weight Base

Input: A matroid $M = (E, I)$ is given as an input as an oracle and a weight function $W : E \to \mathbb{R}$.
Output: Find the maximum weight base of the matroid

begin
  Assume $w(1) \geq \cdots \geq w(n)$
  $S \leftarrow \emptyset$
  $I \leftarrow \{S\}$
  for $i = 1$ *to* n **do**
    if $S + i \in I$ **then**
      $S \leftarrow S + i$
  return S

2.5.2 Correctness Analysis and Characterization

Theorem 2.5.1

The above algorithm outputs a maximum weight base iff M is a matroid

Proof: $\Leftarrow$:
Let M be a matroid. We will prove that this greedy algorithm works by inducting on i. At any iteration i we need to prove the following claim:

Claim: At any iteration i there is a max weight base B_i such that $S_i \subseteq B_i$ and $B_i \setminus S_i \subseteq \{i+1,\dots,n\}$.
Proof: Base case: $S = \emptyset$. So for base case the statement is true trivially. Assume that the statement is true up to $(i-1)$ iterations.

Now $S_{i-1} \subseteq B_{i-1}$ where B_{i-1} is a maximum weight base and $B_{i-1} - S_{i-1} \subseteq \{i,\dots,n\}$. Now three cases arise:

Case 1: If $i \in B_{i-1}$ then $S_{i-1} + i \subseteq B_{i-1}$. Therefore $S_{i-1} + i$ is independent. So now $B_i = B_{i-1}$ and $S_i = S_{i-1} + i$ and $B_i - S_i \subseteq \{i+1,\dots,n\}$.

Case 2: If $i \notin B_{i-1}$ and $S_{i-1} + i \notin I$. Then $S_i = S_{i-1}$ and $B_i = B_{i-1}$. And $B_i - S_i \subseteq \{i+1,\dots,n\}$.

Case 3: If $i \notin B_{i-1}$ but $S_{i-1} + i \in I$. Then $S_i = S_{i-1} + i$. Now S_i can be extended to a B' by adding all but one element of B_{i-1}. So $|B'| = |B_{i-1}|$. Let the element which is not added is $j \in B_{i-1}$. So $B' = B_{i-1} + i - j$.

$$wt(B') = Wt(B_{i-1}) - wt() + wt(i)$$

But we have $wt(i) \geq wt(j)$. So $wt(B') \geq wt(B_{i-1})$. Now since B_{i-1} has maximum weight we have $wt(B') = wt(B_{i-1})$. Then our $B_i = B'$. So $B_i - S_i \subseteq \{i+1,\dots,n\}$.

Hence the claim is true for the ith stage as well. Therefore the claim is true.

■

Therefore using the claim, after the algorithm finished we have no elements left to check, so the current set has the maximum weight which is also an independent set. So the algorithm successfully returns a maximum weight base.
$\Rightarrow$:
Assume M is not a matroid.

■

2.6 Some Matroid Properties

2.6.1 Strong Base Exchange Property

2.6.2 Exchange Graph of a Matroid wrt $S \in I$

Chapter 3

Perfect Matching Polytope

Chapter 4

Bipartite Perfect Matching

Chapter 5

Fractional Matroid Matching

Fractional Matroid Matchings generalizes the case for Matroid Matching or Matroid Parity problem with allowing fractional solutions for the polytope which we will show below. We start with the same kind of state like Matroid Parity Problem

5.1 Fractional Matroid Matchings Polytope

Let $M = (E, \mathcal{I})$ is a matroid with ground set E of even cardinality and with elements E is partitioned into lines or pairs. Let L is the set of lines. Let $r : \mathcal{P}(E) \to \mathbb{Z}$ be the rank function and $sp : \mathcal{P}(E) \to \mathcal{P}(E)$ be the span function. Assume that $\forall\, l \in L, r(L) = 2$. With this setting (same as matroid parity problem) we now define the polytope following [VV92]

Definition 5.1.1: Fractional Matroid Matching Polytope

Let $\mathscr{L}$ denote the lattice of flats in M wtih $S_1 \wedge S_2 = S_1 \cap S_2$ and $S_1 \vee S_2 = sp(S_1 \cup S_2)$ and for each line $l \in L$ let $a_l : \mathscr{L} \to \{0, 1, 2\}$ be the function $a_l(S) = r(sp(l) \cap S)$. Now for any $S \in \mathscr{L}$ and $x \in \mathbb{R}_+^{|L|}$ let $a(S) \cdot x$ denote the vector $(a(S) \cdot x)_l = a_l(S) x_l$ for any $l \in L$. Then the set

$$FP(M) = \{x \in \mathbb{R}_+^{|L|} \mid : a(S) \cdot x \le r(S) \text{ for each } S \in \mathscr{L}\}$$

is fractional matroid matching polytope for M and each vector $x \in FP(M)$ is called a fractional matroid matching.

We take $|L| = m$ to imply that originally the ground set has $2m$ elements. Now we can also allow x to be from $\mathbb{R}^m$, not restricting only to positive vectors. This polytope is a subset of $[0, 1]^m$. We will explain the setting wtih the following example:

Example 5.1

Consider the matroid M with ground set

$$E = \{a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2\}$$

where every 4 element subset of E is a base except these 4 sets

$$\{a_1, a_2, b_1, b_2\}, \quad \{a_1, a_2, c_1, c_2\}, \quad \{a_1, a_2, d_1, d_2\},$$
$$\{b_1, b_2, c_1, c_2\}, \quad \{b_1, b_2, d_1, d_2\}, \quad \{c_1, c_2, d_1, d_2\}$$

Now the lines are defined to be

$$l_1 = \{a_1, a_2\} \qquad l_2 = \{b_1, b_2\}, \qquad l_3 = \{c_1, c_2\}, \qquad l_4 = \{d_1, d_2\}$$

Now the flats of M are empty set, individual elements, every pair of elements, set consists of one element from each of three lines, pair of line and E. Hence $FP(M)$ is the set of $x \in \mathbb{R}_+^{|L|}$ satisfying

$$\begin{array}{ccc} 2x_1 + 2x_2 \leq 3 & 2x_1 + 2x_3 \leq 3 & 2x_1 + 2x_4 \leq 3 \\ 2x_2 + 2x_3 \leq 3 & 2x_2 + 2x_4 \leq 3 & 2x_3 + 2x_4 \leq 3 \end{array}$$

$$2x_1 + 2x_2 + 2x_3 + 2x_4 \leq 4$$

$$2x_i \leq 2 \quad \text{for each } i \in [4]$$

Now the we show the theorem Theorem 5.1.1 which states that the fractional matroid matching polytope arises as a linear relaxation of the matroid matching problem.

Theorem 5.1.1 [VV92, Theoerm 2.1]
An integer vector $x \in \mathbb{R}_+^m$ is the incidence vector of a matroid matching iff x is a fractional matroid matching.

You can clearly see this theorem by comparing the Matroid Matching Polytope and Fractional Matroid Matching Polytope so we are omitting the proof.

Theorem 5.1.2 [GP13, Theorem 1]
The vertices of the fractional matroid matching are half-integral

5.1.1 Weighted Fractional Matroid Matching

Definition 5.1.2: Weighted Fractional Matroid Matching Problem

It is to find a fractional matroid matching x that maximizes $w \cdot x$ for a non-negative weight assignment $w : L \to \mathbb{Z}_+$.

For plain Fractional Linear Matroid Matching Problem we need to find a fractional matroid matching x which maximizes the size i.e. L_1 norm of x which is $\sum\limits_{l \in L} |x_l|$.

Gijswijt and Pap in [GP13] gave a polynomial time algorithm for weighted fractional linear matroid matching. They also gave the following characterization for maximizing face of the polytope with respect to a weight function.

Theorem 5.1.3 [GP13, Prood of Theorem 1]
Let $L = \{l_1, \ldots, l_m\}$ be a set of lines with $l_i \subseteq \mathbb{F}^n$ and $w : L \to \mathbb{Z}$ be a weight assignment on L. Let F denote the set of fractional linear matroid matchings maximizing and $S \subseteq [m]$ such that every $x \in F$ has $y_e = 0$ for all $e \in S$. Then for some $k \leq n$, $\exists$ a $k \times m$ matrix D_F and $b_F \in \mathbb{Z}^k$ such that

- $D_F \in \{0, 1, 2\}^{k \times m}$
- The sum of entries in any column of D_F is exactly 2
- A fractional matroid matching x is in F iff $y_e = 0$ for $e \in S$ and $D_F x = b_F$.

5.2 Isolating Weight Assignment for Fractional Matroid Matching

In this section we will describe how we can construct an isolating weight assignment for fractional matroid matching with just the number of lines as input.

Now for a face F of a polytope, let $\mathcal{L}_F$ denote the lattice

$$\mathcal{L}_F = \{v \in \mathbb{Z}^m \mid v = \alpha(x_1 - x_2) \text{ for some } x_1, x_2 \in F \text{ and } \alpha \in \mathbb{R}\}$$

and $\lambda(\mathcal{L}_F)$ denote the length of the shortest vector of $\mathcal{L}_F$. Hence $\mathcal{L}_F$ consistes of all integral vectors parallel to the face F.

Now by Theorem 5.1.3 the face maximizing the size is described by the equation $D_F x = b_F$ where $D_F \in \{0,1,2\}^{k\times m}$ with column sum 2. Hence $\mathcal{L}_F$ is exactly the set of integral vectors in the null space of D_F. Therefore

$$\mathcal{L}_F = \{v \in \mathbb{Z}^m \mid D_F v = 0\}$$

So we will prove that the number of vectors in $\mathcal{L}_F$ with size less than twice the length of shortest vector is polynomially bounded in Subsection 1.2.2.

First we will show how we can interpret D_F an incidence matrix for a graph instead of general matrix. $D_F \in \{0,1,2\}^{k\times m}$ where every column sum is 2. Hence we can thing of a graph G_D with vertex set $[k]$ and the m edges defined as follows: for every $e \in [m]$ the e-th edge of G_D is drawn between the vertices $s,t \in [k]$ if $D_F[s,e] = D_F[t,e] = 1$ and e-th edge is a self look on the vertex $s \in [k]$ if $D_F[s,e] = 2$.

5.2.1 Alternating Circuits

First we will define some elements called alternating indicator vetors and alternating circuits following the [ST17] for the proof of Theorem 5.2.3.

Definition 5.2.1: Alternating Indicator Vector & Alternating Circuit

Let $C = v_0 \xrightarrow{e_0} v_1 \xrightarrow{e_1} v_2 \cdots \xrightarrow{e_{k-2}} v_{k-1} \xrightarrow{e_{k-1}} v_0$ be a closed walk of even length in a multigraph G with loops. Then the *Alternating Indicator Vector* of C denoted by $(\pm\mathbb{1})_C$ is the vector

$$(\pm\mathbb{1})_C \coloneqq \sum_{i=0}^{k-1} (-1)^i \mathbb{1}_{e_i}$$

C is called *Altenating Circuit* if its alternating indicator vector is nonzero

We can use the parity for $(-1)^i$ as direction of movement in C. So a closed walk C is a alternating circuti if there exists at least one edge $e \in C$ for which C has moved through e more time in one direction than the other.

Observation. $|(\pm\mathbb{1})_C| \leq |C|$ *for any even length closed walk* C.

Now we will prove a property for all alternating circuits in G_D

Lemma 5.2.1 [GOR24, Proof of Claim 1, Proof of Theorem 3.4]
For any alternating circuit C we have $D_F \cdot (\pm\mathbb{1})_C = 0$

Proof: Suppose $C = v_0 \xrightarrow{e_0} v_1 \xrightarrow{e_1} v_2 \cdots \xrightarrow{e_{k-2}} v_{k-1} \xrightarrow{e_{k-1}} v_0$. We denote the i-th column of D_F is denoted by D_i. Now

$$D_i \cdot (\pm\mathbb{1})_C = \sum_{j=0}^{k-1} (-1)^j D_F[i,e_j] = \sum_{e_j : i \in e_j \in C} (-1)^j D_F[i,e_j]$$

Hence only the edges in C which are incident on i contributes to the above sum. Therefore the whole sum is partitioned into distinct subparts of walks where each subpart is of the form

$$v_s \xrightarrow{e_s} \underbrace{i \xrightarrow{e_{s+1}} i \xrightarrow{e_{s+2}} \cdots \xrightarrow{e_{s+k-1}} i}_{k \text{ times}} \xrightarrow{e_{s+k}} v_t \qquad \text{where } v_s, v_t \neq i$$

i.e. the part starts from a vertex not equal to i then goes to i and after looping in i for some times the walk goes to another vertex not equal to i. We will show that for each of these parts the contribution to the sum is 0.

Now for such a subpart their contribution to the sum is

$$\delta = (-1)^s + 2\sum_{j=1}^{k-1} (-1)^{s+i} + (-1)^{s+k}$$

We will analyze case wise:

Case 1: k is odd: Then $k-1$ is even. Hence

$$\sum_{j=1}^{k-1}(-1)^{s+i} = (-1)^s\sum_{j=1}^{k-1}(-1)^i = 0$$

Therefore

$$\delta = (-1)^s + (-1)^{s+k} = (-1)^s[1+(-1)^k] = 0$$

as k is odd.

Case 2: k is even: Then $k-2$ is even. Hence

$$\sum_{j=1}^{k-1}(-1)^{s+i} = \left[\sum_{j=1}^{k-2}(-1)^{s+i}\right] + (-1)^{s+k-1} = (-1)^{s+k-1}$$

Hence

$$\delta = (-1)^s + 2(-1)^{s+k-1} + (-1)^{s+k} = (-1)^s[1+2(-1)^{k-1}+(-1)^k] = 0$$

Therefore we showed that for each such parts their contribution to the sum is 0. Therefore the total sum is 0 i.e. $D_i \cdot (\pm\mathbb{1})_C = 0$. Since this is true for all $i \in [k]$ we have $D_F \cdot (\pm\mathbb{1})_C = 0$. ■

Now we will show that any vector in the lattice $\mathcal{L}_F$ can be decomposed into finite sum of alternating indicator vectors of alternating circuits with the property that for each such alternating circuit C, $|(\pm\mathbb{1})_C| = |C|$. Before that we introduce a relation between two vectors in $\mathbb{R}^m$ this will come in handy for the decomposition.

Definition 5.2.2: Conformal

For $x, y \in \mathbb{R}^m$, x is said to be conformal to y if $x_i y_i \geq 0$ and $|x_i| \leq |y_i|\ \forall\, i \in [m]$ and it is denoted by $x \sqsubseteq y$

Lemma 5.2.2 [GOR24, Claim 1, Proof of Theorem 3.4]
For any $x \in \mathcal{L}_F$, $\exists$ alternating circuits $C_1, C_2, \ldots, C_t$ in G_D such that

$$x = \sum_{i=1}^{t}(\pm\mathbb{1})_{C_i}$$

where $\forall\, i \in [t]$, $(\pm\mathbb{1})_{C_i} \sqsubseteq x$ and $|(\pm\mathbb{1})_{C_i}| = |C_i|$.

Proof: We will decompose a given x into alternating indicator vectors by the following iterative algorithm:

Algorithm 2: Decomposition of a Lattice Vector

Input: $x \in \mathcal{L}_F$
Output: $\mathcal{Y} = \{y_i\}$ where $|\mathcal{Y}| < \infty$ and $\forall\, y_i \in \mathcal{Y}$ are alternating indicator vectors

1 **begin**
2   **while** $|x| \neq 0$ **do**
3     $y \longleftarrow 0, j \longleftarrow 1$
4   Let $e_0 \in [m]$ such that $x_{e_0} > 0$
5     $y_{e_0} \longleftarrow 1$ and let e_0-th edge in G_D be $\{v_0, v_1\}$
6   **while** *True* **do**
7   **if** $\exists\, e \in [m]$ *such that* e*-th edge is* $\{v_j, u\}$, $|x_e| > |y_e|$ *and* $(-1)^j x_e > 0$ **then**
8         $y_e \longleftarrow y_e + (-1)^j, e_j \longleftarrow e$
9         $v_{j+1} \longleftarrow u$
10         $j \longleftarrow j + 1$
11   **else**
12   **return** $x \notin \mathcal{L}_F$
13   **if** $j \equiv 0 \pmod 2$ *and* $v_j = v_0$ **then**
14         $x \longleftarrow x - y$
15   **return** y and exit inner while loop

Now suppose x denote the vector at some iteration of the outer while loop. Now forall $e \in [m]$, let at jth and lth iteration of the inner while loop $(-1)^j$ and $(-1)^l$ are added to y_e respectively. Now both j and l has same parity because since the edge is not changing both times $(-1)^j x_e > 0$ and $(-1)^l x_e > 0$ has to be satisfies. Therefore we get j and l have same parity. Hence for a full run of the inner while loop for any $e \in [m]$ everytime y_e is changed the same $(-1)^j$ is added. Hence whenever y_e is changed $|y_e|$ is increased. Therefore $|y|$ increases for each iteration of the inner while loop. But $|y|$ cannot exceed $|x|$ since as the addition step works when $\exists\, e \in [m]$ with $|x_e| > |y_e|$ and $(-1)^j x_e > 0$, after addition $|y_e + (-1)^j| \leq |y_e| + 1 \leq |x|$. So if we start with $|y| \leq |x|$ after addition with each iteration of inner while loop we still have $|y| \leq |x|$. Also since for every such edge we add $(-1)^j$ to y_e when $(-1)^j x_e > 0$ and $|x_e| > |y_e|$. So by the first condition we have y_e and x_e has the same sign i.e. $x_e y_e > 0$. And we also have $|x_e| \geq |y_e|$. Therefore we have $y \sqsubseteq x$.

Now since we start the algorithm by adding 1 to y_{e_0} at Line 5 we have initially $|y| > 0$ and afterwards with each iteration of the inner while loop $|y|$ increases so we have $|y| > 0$.Now if the current iteration of the inner while loop ends at Line 13 then we get a closed walk C since in each iteration of the inner while loop the algorithm follows a walk in G_D. So C is an alternating circuit and we have $|y| = |C|$. Now since $y \sqsubseteq x$ we have $|x - y| = |x| - |y| < |x|$. Since C is alternating circuit by Lemma 5.2.1 we have $y \in \mathcal{L}_F$. Hence we have $x - y \in \mathcal{L}_F$.

Now all that remains is to show that the algorithm never goes to Line 11 if $x \in \mathcal{L}_F$. The only reason the algorithm can go to Line 11 if at some iteration of the inner while loop can not find an edge $e \in [m]$ such that $|x_e| > |y_e|$ and $(-1)^j x_e > 0$ where $e \in \delta(v_j)$ where $\delta(v_j)$ denotes te set of edges incident on v_j. Suppose at this point we have the walk

$$\mathcal{P} = v_0 \xrightarrow{e_0} v_1 \xrightarrow{e_1} \cdots \xrightarrow{e_{j-1}} v_j$$

Now by following the proof of Lemma 5.2.1 we have that $D_i y = 0$ forall $i \in [k]$ except $i \notin \{v_0.v_j\}$.

Claim: If j is odd then $D_{v_j} y > 0$ and if j is even then $D_{v_j} y < 0$.
Proof: We will prove the case for j is odd. The even case will follow similarly. Since j is odd we have $j - 1$ even. Therefore $x_{e_{j-1}} > 0$ as $(-1)^{j-1} x_{e_{j-1}} > 0$. Therefore for both e_0 and e_{j-1}, $x_{e_0}, x_{e_{j-1}}$ positive. Now take the partitions as described in the proof of Lemma 5.2.1. Now we analyze case wise:
Case 1: $v_0 \neq v_j$**:** Then only the partition containing v_{e_j} might contribute something nonzero because other partitions contributes 0 to the sum $D_{v_j} y$. Let the partition be

$$v_s \xrightarrow{e_{j-k-1}} \underbrace{v_j \xrightarrow{e_{j-k}} v_j \xrightarrow{e_{j-k+1}} \cdots \xrightarrow{e_{j-1}} v_j}_{k+1 \text{ times}} \qquad \text{where } v_s \neq v_j$$

Now k can not be more than 1 since if e_{j-1} and e_{j-2} are both loops in v_j then for e_{j-1}, $x_{e_{j-1}} > 0$ but $x_{e_{j-2}} < 0$ but $e_{j-1} = e_{j-2} \implies 0 < x_{e_{j-1}} = x_{e_{j-2}} < 0$. Then this partition contributes $\delta = (-1)^{j-2} + 2(-1)^{j-1} = -1 + 2 > 0$ to $D_{v_j} y$ if there is a loop and if there is no loop then it also contributes 1 as then the contribution will be only $(-1)^{j-1} = 1 > 0$.

Case 2: $v_0 = v_j$**:** If v_0 and v_j is in same partition i.e. $j = 1$ then the it contributes $(-1)^0 = 1 > 0$ to the sum. Otherwise v_0 is in different partition. By the above case we know that the value contributed by the partition containing v_j is 1. So we have to only find the contribution by the partition containing v_0. Again by same logic as the above case from v_0 at most one loop starting from v_0 and then it moves to some other vertex. Hence the sum contributed is $2(-1)^0 + (-1)^1 = 2 - 1 = 1$ if there is a loop and if there is no loop then it contributes $(-1)^0 = 1$. Hence in both cases the contribution of the partition containing v_0 to the sum $D_{v_j}y$ is 1. Hence we have $D_{v_j}y > 0$.

Hence by above we get that if j is odd then the value $D_{v_j}y > 0$. Similarly following the same process in the case of j is even we get $D_{v_j}y < 0$. ■

Hence by the lemma we have that if j is odd $D_{v_j}y > 0$ and if j is even then $D_{v_j}y < 0$. So WLOG suppose j is odd. Now define

$$\text{supp}^+(x) = \{i \in [m] \mid x_i > 0\} \qquad \text{and} \qquad \text{supp}^-(x) = \{i \in [m] \mid x_i < 0\}$$

Then we have

$$\begin{aligned} 0 = D_{v_j}x &= \sum_{e \in \text{supp}^+(x) \cap \delta(v_j)} D_F[v_j, e] \cdot |x_e| - \sum_{e \in \text{supp}^-(x) \cap \delta(v_j)} D_F[v_j, e] \cdot |x_e| \\ 0 < D_{v_j}y &= \sum_{e \in \text{supp}^+(x) \cap \delta(v_j)} D_F[v_j, e] \cdot |y_e| - \sum_{e \in \text{supp}^-(x) \cap \delta(v_j)} D_F[v_j, e] \cdot |y_e| \end{aligned}$$

Now be construction of y we have $y \sqsubseteq x \implies |x_e| \geq |y_e|, x_e y_e > 0 \forall\, e \in [m]$. The algorithm stopped at Line 13 since $\nexists\, e \in \delta(v_j)$ such that $|x_e| > |y_e|$ and $(-1)^j x_e = -x_e > 0$. Therefore $|x_e| = |y_e|$ for all $e \in \text{supp}^+(x) \cap \delta(v_j)$ otherwise the algorithm would have proceeded one more step and $\text{supp}^-(x) \cap \delta(v_j) = \text{supp}^-(y) \cap \delta(v_j)$ since for any $e \in [m]$, $x_e y_e > 0$. Therefore $\text{supp}^+(x) \cap \delta(v_j) = \text{supp}^+(y) \cap \delta(v_j)$. Now for all $e \in \text{supp}^+(x) \cap \delta(v_j)$ we have $|x_e| \geq |y_e|$. Hence we have

$$0 = D_{v_j}y \geq D_{v_j}y > 0$$

It is a contradiction. Hence if $x \in \mathcal{L}_F$ the algorithm never goes to Line 13. And therefore the algorithm successfully decomposes x into sum of alternating circuits. ■

5.2.2 Bounding vectors in $\mathcal{L}_F$ with Small Size

Theorem 5.2.3 [GOR24]
Let $D \in \{0, 1, 2\}^{p \times m}$ be a matrix such that the sum of entries of each column equals 2. Let $\mathcal{L}_D$ denote the lattice $\{v \in \mathbb{Z}^m \mid Dv = 0\}$. Then it holds that

$$|\{v \in \mathcal{L}_D \mid |v| < 2\lambda(\mathcal{L}_D)\}| \leq m^{O(1)}$$

Proof: For the given D consider the graph G_D obtained from D as explained at the start of Section 1.2. to show that the number of vectors in $\mathcal{L}_D$ with size less than twice the size of shortest vector in $\mathcal{L}_D$ we will show that for any such lattice vector there is only one alternating circuit in the decomposition of the vector in Lemma 5.2.2.

Claim: Any lattice vector $x \in \mathcal{L}_D$ with $|x| < 2\lambda(\mathcal{L}_D)$ is an alternating vector $(\pm\mathbb{1})_C$ of some alternating circuit C in G_D such that $|x| = |C|$

Proof: Suppose the contrary. Since $x \in \mathcal{L}_D$ by Lemma 5.2.2 $\exists\, C_1, \ldots, C_t$ with $t \geq 2$ such that $x = \sum_{i=1}^{t} (\pm\mathbb{1})_C$ with $(\pm\mathbb{1})_C \sqsubseteq x$ and $|(\pm\mathbb{1})_{C_i}| = |C_i|$ for all $i \in [t]$. Then we have

$$|x| = \sum_{i=1}^{t} |(\pm\mathbb{1})_{C_i}| \geq t\lambda(\mathcal{L}_D) \geq 2\lambda(\mathcal{L}_D)$$

which is a contradiction since we assumed that $|x| < 2\lambda(\mathcal{L}_D)$. Hence $t = 1$ i.e. $x = (\pm\mathbb{1})_C$ for some alternating circuit C with $|x| = |C|$. ■

Hence the Claim implies that $\lambda(\mathcal{L}_D)$ is equal to the size of the smallest alternating circuit of G_D. And it also implies that we only need to bound the number of alternating indicator vectors that correspond to alternating circuit of size at most $2\lambda(\mathcal{L}_D)$ to prove the lemma. For that by the Theorem 5.2.4 we get that the number of such alternating indicator vectors are polynomially bounded by m. ■

Theorem 5.2.4 [ST17, Lemma 5.4]

Let G be a graph on n vertices such that the size of the smallest alternating circuit λ. Then the cardinality of the set

$$\{(\pm\mathbb{1})_C \mid C \text{ is an alternating circuit in } G \text{ of size at most } 2\lambda\}$$

is at most n^{17}.

Proof: content... ■

5.2.3 Algorithm for Finding Isolating Weight Assignment

With this theorem we have

Theorem 5.2.5 [GTV21, Theorem 2.5]

Let k be a positive integer and $P \subseteq \mathbb{R}^m$ a polytope such that its extreme ppoints are in $\left\{0, \frac{1}{k}, \frac{2}{k}, \ldots, 1\right\}^m$ and there exists a constant $c > 1$ with

$$|\{v \in \mathcal{L}_F \colon |v| < c\lambda(\mathcal{L}_F)\}| \leq m^{O(1)}$$

for any face F of P. Then there exists an algorithm that, given k and m, outputs a set $\mathcal{W} \subseteq \mathbb{Z}^m$ of $m^{O(\log km)}$ weight assignments with weights bounded by $m^{O(\log km)}$ such that there exists at least one $w \in \mathcal{W}$ that is isolating for P, in time $polylog(km)$ using $m^{O(\log km)}$ many parallel processors.

Using this we finally have an algorithm for isolating a fractional matroid matching polytope:

Theorem 5.2.6 [GOR24, Theorem 3.1]

There exists an algorithm that given $m \in \mathbb{Z}_+$ outputs a set $\mathcal{W} \subseteq \mathbb{Z}_+^m$ of $m^{O(\log m)}$ weight assignments with weights bounded by $m^{O(\log m)}$ such that, for any fractional matroid matching polytope P of m lines, there exists at least one $w \in \mathcal{W}$ that is isolating for P, in time $polylog(m)$ usign $m^{O(\log m)}$ many parallel processors.

Chapter 6

Bibliography

[GOR24] Rohit Gurjar, Taihei Oki, and Roshan Raj. Fractional Linear Matroid Matching is in quasi-NC. 2024.

[GP13] Dion Gijswijt and Gyula Pap. An algorithm for weighted fractional matroid matching. *Journal of Combinatorial Theory, Series B*, 103(4):509–520, July 2013.

[GTV21] Rohit Gurjar, Thomas Thierauf, and Nisheeth K. Vishnoi. Isolating a Vertex via Lattices: Polytopes with Totally Unimodular Faces. *SIAM Journal on Computing*, 50(2):636–661, January 2021.

[ST17] Ola Svensson and Jakub Tarnawski. The Matching Problem in General Graphs Is in quasi-NC. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, October 2017.

[VV92] John H Vande Vate. Fractional Matroid Matchings. *Journal of Combinatorial Theory, Series B*, 55(1):133–145, May 1992.