EE595A – Submodular functions, their optimization and applications – Spring 2011

Prof. Jeff Bilmes

University of Washington, Seattle
Department of Electrical Engineering

Spring Quarter, 2011
http://ssli.ee.washington.edu/~bilmes/ee595a_spring_2011/

Lecture 9 - April 29th, 2011

Announcements

- HW2 should be hopefully ready by this weekend (I'll send email when ready).
- On Final projects. **One** single page final project proposals (revision one) are due next Friday (one week from today) at 6:00pm.
- Again, all submissions must be done electronically, via our drop box. See the link `https://catalyst.uw.edu/collectit/dropbox/bilmes/14888`, or look at the homework on the web page.
- Email me and/or stop by office hours for ideas. The proposals next Friday are non-binding (you can change your mind later) but you should start thinking about project proposals now.
- Ideal proposal would lead to a NIPS paper in June and be related to submodularity.

A polymatroid function's polyhedron is a polymatroid.

Theorem 2.1

Let f be a polymatroid function defined on subsets of E. For any $x \in \mathbb{R}_+^E$, and any P_f-basis y^x of x, we have that the component sum of y is

$$\max\left(y(E) : y \leq x, y \in P_f\right) = y^x(E) = \min\left(x(A) + f(E \setminus A) : A \subseteq E\right) \tag{1}$$

A polymatroid function's polyhedron is a polymatroid.

Theorem 2.1

Let f be a polymatroid function defined on subsets of E. For any $x \in \mathbb{R}_+^E$, and any P_f-basis y^x of x, we have that the component sum of y is

$$\max\left(y(E) : y \leq x, y \in P_f\right) = y^x(E) = \min\left(x(A) + f(E \setminus A) : A \subseteq E\right) \tag{1}$$

There, as we will see, are a number of consequences of this theorem (other than that P_f is a polymatroid).

Matroid case

- Considering the above theorem, the matroid case is now a special case, where we have that:

Corollary 2.2

We have that:

$$\max \{y(E) : y \in P_{ind.\ set}(M), y \leq x\} = \min \{r_M(A) + x(E \setminus A) : A \subseteq E\} \tag{2}$$

where r_M is the matroid rank function of some matroid.

Matroid from submodular function

Theorem 2.3

Given integral polymatroid function f, let $(E, \mathcal{F})$ be a set system with ground set E and set of subsets $\mathcal{F}$ such that

$$\forall J \in F, \forall \emptyset \subset S \subseteq J, |S| \leq f(S) \tag{3}$$

Then $M = (E, \mathcal{F})$ is a matroid.

Proof.

Exercise □

And its rank function is Exercise.

Most violated inequality problem

- Consider

$$P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \tag{4}$$

Most violated inequality problem

- Consider
$$P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \tag{4}$$
- We saw before that $P_r = P_{\text{ind. set}}$.

Most violated inequality problem

- Consider

$$P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \tag{4}$$

- We saw before that $P_r = P_{\text{ind. set}}$.
- Suppose we have any $x \in \mathbb{R}^E_+$ such that $x \notin P_r$.

Most violated inequality problem

- Consider
$$P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \tag{4}$$
- We saw before that $P_r = P_{\text{ind. set}}$.
- Suppose we have any $x \in \mathbb{R}^E_+$ such that $x \notin P_r$.
- The most violated inequality when x is considered w.r.t. P_r corresponds to the set A that maximizes $x(A) - r_M(A)$, i.e., $\max \{x(A) - r_M(A) : A \subseteq E\}$.

Most violated inequality problem

- Consider
$$P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \tag{4}$$
- We saw before that $P_r = P_{\text{ind. set}}$.
- Suppose we have any $x \in \mathbb{R}_+^E$ such that $x \notin P_r$.
- The most violated inequality when x is considered w.r.t. P_r corresponds to the set A that maximizes $x(A) - r_M(A)$, i.e., $\max \{x(A) - r_M(A) : A \subseteq E\}$.
- This corresponds to $\min \{r_M(A) + x(E \setminus A) : A \subseteq E\}$ since x is modular and $x(E \setminus A) = x(E) - x(A)$.

Most violated inequality problem

- Consider
$$P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \tag{4}$$
- We saw before that $P_r = P_{\text{ind. set}}$.
- Suppose we have any $x \in \mathbb{R}_+^E$ such that $x \notin P_r$.
- The most violated inequality when x is considered w.r.t. P_r corresponds to the set A that maximizes $x(A) - r_M(A)$, i.e., $\max \{x(A) - r_M(A) : A \subseteq E\}$.
- This corresponds to $\min \{r_M(A) + x(E \setminus A) : A \subseteq E\}$ since x is modular and $x(E \setminus A) = x(E) - x(A)$.
- More importantly, $\min \{r_M(A) + x(E \setminus A) : A \subseteq E\}$ a form of submodular function minimization, namely $\min \{r_M(A) - x(A) : A \subseteq E\}$ for a submodular function consisting of a difference of matroid rank and modular (so no longer nec. monotone, nor positive).

Problem To Solve

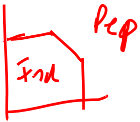
In particular, we will solve the following problem:

- Given a matroid $M = (E, \mathcal{I})$ along with an independence testing oracle (i.e., for any $A \subseteq E$, tells us if $A \in \mathcal{I}$ or not), and a vector $x \in \mathcal{R}_+^E$;

Problem To Solve

In particular, we will solve the following problem:

- Given a matroid $M = (E, \mathcal{I})$ along with an independence testing oracle (i.e., for any $A \subseteq E$, tells us if $A \in \mathcal{I}$ or not), and a vector $x \in \mathcal{R}_+^E$;
- find: a maximizing $y \in P_{\text{ind. set}}$ with $y \leq x$, and moreover (as a byproduct of the algorithm), express y as a convex combination of incidence vectors of independent sets in M, and also return a set $A \subseteq E$ that satisfies $y(E) = r_M(A) + x(E \setminus A)$. *Of course, for any such y we must have that $y(E) \leq r(A) + x(E \setminus A)$.*



Problem To Solve

In particular, we will solve the following problem:

- Given a matroid $M = (E, \mathcal{I})$ along with an independence testing oracle (i.e., for any $A \subseteq E$, tells us if $A \in \mathcal{I}$ or not), and a vector $x \in \mathcal{R}_+^E$;
- find: a maximizing $y \in P_{\text{ind. set}}$ with $y \leq x$, and moreover (as a byproduct of the algorithm), express y as a convex combination of incidence vectors of independent sets in M, and also return a set $A \subseteq E$ that satisfies $y(E) = r_M(A) + x(E \setminus A)$. *Of course, for any such y we must have that $y(E) \leq r(A) + x(E \setminus A)$.*
- By the above theorem, the existence of such an A will certify that $x(E)$ is maximal in $P_{\text{ind. set}}$, minimal in terms of $f(A) = r_M(A) - x(A)$, and thus most violated in terms of A.

Problem To Solve

In particular, we will solve the following problem:

- Given a matroid $M = (E, \mathcal{I})$ along with an independence testing oracle (i.e., for any $A \subseteq E$, tells us if $A \in \mathcal{I}$ or not), and a vector $x \in \mathcal{R}_+^E$;
- find: a maximizing $y \in P_{\text{ind. set}}$ with $y \leq x$, and moreover (as a byproduct of the algorithm), express y as a convex combination of incidence vectors of independent sets in M, and also return a set $A \subseteq E$ that satisfies $y(E) = r_M(A) + x(E \setminus A)$. *Of course, for any such y we must have that $y(E) \leq r(A) + x(E \setminus A)$.*
- By the above theorem, the existence of such an A will certify that $x(E)$ is maximal in $P_{\text{ind. set}}$, minimal in terms of $f(A) = r_M(A) - x(A)$, and thus most violated in terms of A.
- This can also be used to test membership in $P_{\text{ind. set}}$ (i.e., if $y = x$) depending on the sign of f.

Problem To Solve

In particular, we will solve the following problem:

- Given a matroid $M = (E, \mathcal{I})$ along with an independence testing oracle (i.e., for any $A \subseteq E$, tells us if $A \in \mathcal{I}$ or not), and a vector $x \in \mathcal{R}_+^E$;
- find: a maximizing $y \in P_{\text{ind. set}}$ with $y \leq x$, and moreover (as a byproduct of the algorithm), express y as a convex combination of incidence vectors of independent sets in M, and also return a set $A \subseteq E$ that satisfies $y(E) = r_M(A) + x(E \setminus A)$. *Of course, for any such y we must have that $y(E) \leq r(A) + x(E \setminus A)$.*
- By the above theorem, the existence of such an A will certify that $x(E)$ is maximal in $P_{\text{ind. set}}$, minimal in terms of $f(A) = r_M(A) - x(A)$, and thus most violated in terms of A.
- This can also be used to test membership in $P_{\text{ind. set}}$ (i.e., if $y = x$) depending on the sign of f.
- This will also run in polynomial time.

Idea of the algorithm

- We build up y from the ground up.

Idea of the algorithm

- We build up y from the ground up.
- We keep a family of independent sets $(I_i : i \in J)$ and coefficients $(\lambda_i : i \in J)$ such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.

Idea of the algorithm

- We build up y from the ground up.
- We keep a family of independent sets $(I_i : i \in J)$ and coefficients $(\lambda_i : i \in J)$ such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- We gradually build up y by adding new independent sets (and augmenting J), adding to the existing independent sets, and adjusting coefficients.

Idea of the algorithm

- We build up y from the ground up.
- We keep a family of independent sets $(I_i : i \in J)$ and coefficients $(\lambda_i : i \in J)$ such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- We gradually build up y by adding new independent sets (and augmenting J), adding to the existing independent sets, and adjusting coefficients.
- and the way these additions are done is via solutions to a max-flow problem in an associated flow-graph (which we'll describe).

Idea of the algorithm

- We build up y from the ground up.
- We keep a family of independent sets $(I_i : i \in J)$ and coefficients $(\lambda_i : i \in J)$ such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- We gradually build up y by adding new independent sets (and augmenting J), adding to the existing independent sets, and adjusting coefficients.
- and the way these additions are done is via solutions to a max-flow problem in an associated flow-graph (which we'll describe).
- Each update will, of course, ensure that $y \in P_{\text{ind. set}}$, but also we'll keep $y \leq x$.

Idea of the algorithm

- We build up y from the ground up.
- We keep a family of independent sets $(I_i : i \in J)$ and coefficients $(\lambda_i : i \in J)$ such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- We gradually build up y by adding new independent sets (and augmenting J), adding to the existing independent sets, and adjusting coefficients.
- and the way these additions are done is via solutions to a max-flow problem in an associated flow-graph (which we'll describe).
- Each update will, of course, ensure that $y \in P_{\text{ind. set}}$, but also we'll keep $y \leq x$.
- It's going to take us a few lectures to fully develop this algorithm, so please keep mind of the overall goal.

Matroid Intersection Algorithm Idea

- Consider two matroids $M_1 = (V, \mathcal{I}_1)$ and $M_2 = (V, \mathcal{I}_2)$ and start with any $I \in \mathcal{I}_1 \cap \mathcal{I}_2$.

Matroid Intersection Algorithm Idea

- Consider two matroids $M_1 = (V, \mathcal{I}_1)$ and $M_2 = (V, \mathcal{I}_2)$ and start with any $I \in \mathcal{I}_1 \cap \mathcal{I}_2$.
- Consider some $v_1 \notin \text{span}_1(I)$, so that $I + v_1 \in \mathcal{I}_1$.

Matroid Intersection Algorithm Idea

- Consider two matroids $M_1 = (V, \mathcal{I}_1)$ and $M_2 = (V, \mathcal{I}_2)$ and start with any $I \in \mathcal{I}_1 \cap \mathcal{I}_2$.
- Consider some $v_1 \notin \text{span}_1(I)$, so that $I + v_1 \in \mathcal{I}_1$.
- If $I + v_1 \in \mathcal{I}_2$, then v_1 is "augmenting", and we can augment I to $I + v_1$ and still be independent in both M_1 and M_2.

Matroid Intersection Algorithm Idea

- Consider two matroids $M_1 = (V, \mathcal{I}_1)$ and $M_2 = (V, \mathcal{I}_2)$ and start with any $I \in \mathcal{I}_1 \cap \mathcal{I}_2$.
- Consider some $v_1 \notin \text{span}_1(I)$, so that $I + v_1 \in \mathcal{I}_1$.
- If $I + v_1 \in \mathcal{I}_2$, then v_1 is "augmenting", and we can augment I to $I + v_1$ and still be independent in both M_1 and M_2.
- If $I + v_1 \notin \mathcal{I}_2$, $\exists C_2(I, v_1)$ a circuit in M_2, and choosing $v_2 \in C_2(I, v_1)$ s.t. $v_2 \neq v_1$ leads to $I + v_1 - v_2$ which (because $\text{span}_2(I) = \text{span}(I + v_1 - v_2)$) is again independent in M_2. It is also independent in M_1.

Matroid Intersection Algorithm Idea

- Consider two matroids $M_1 = (V, \mathcal{I}_1)$ and $M_2 = (V, \mathcal{I}_2)$ and start with any $I \in \mathcal{I}_1 \cap \mathcal{I}_2$.
- Consider some $v_1 \notin \text{span}_1(I)$, so that $I + v_1 \in \mathcal{I}_1$.
- If $I + v_1 \in \mathcal{I}_2$, then v_1 is "augmenting", and we can augment I to $I + v_1$ and still be independent in both M_1 and M_2.
- If $I + v_1 \notin \mathcal{I}_2$, $\exists C_2(I, v_1)$ a circuit in M_2, and choosing $v_2 \in C_2(I, v_1)$ s.t. $v_2 \neq v_1$ leads to $I + v_1 - v_2$ which (because $\text{span}_2(I) = \text{span}(I + v_1 - v_2)$) is again independent in M_2. It is also independent in M_1.
- Next choose a $v_3 \in \text{span}_1(I) - \text{span}_1(I - v_2)$ to recover what was lost in $I \cup \{v_1\}$ when we removed v_2 from it.

Matroid Intersection Algorithm Idea

- Consider two matroids $M_1 = (V, \mathcal{I}_1)$ and $M_2 = (V, \mathcal{I}_2)$ and start with any $I \in \mathcal{I}_1 \cap \mathcal{I}_2$.
- Consider some $v_1 \notin \text{span}_1(I)$, so that $I + v_1 \in \mathcal{I}_1$.
- If $I + v_1 \in \mathcal{I}_2$, then v_1 is "augmenting", and we can augment I to $I + v_1$ and still be independent in both M_1 and M_2.
- If $I + v_1 \notin \mathcal{I}_2$, $\exists C_2(I, v_1)$ a circuit in M_2, and choosing $v_2 \in C_2(I, v_1)$ s.t. $v_2 \neq v_1$ leads to $I + v_1 - v_2$ which (because $\text{span}_2(I) = \text{span}(I + v_1 - v_2)$) is again independent in M_2. It is also independent in M_1.
- Next choose a $v_3 \in \text{span}_1(I) - \text{span}_1(I - v_2)$ to recover what was lost in $I \cup \{v_1\}$ when we removed v_2 from it.
- Then $\text{span}_1(I) = \text{span}_1(I - v_2 + v_3)$.

Matroid Intersection Algorithm Idea

- Consider two matroids $M_1 = (V, \mathcal{I}_1)$ and $M_2 = (V, \mathcal{I}_2)$ and start with any $I \in \mathcal{I}_1 \cap \mathcal{I}_2$.
- Consider some $v_1 \notin \text{span}_1(I)$, so that $I + v_1 \in \mathcal{I}_1$.
- If $I + v_1 \in \mathcal{I}_2$, then v_1 is "augmenting", and we can augment I to $I + v_1$ and still be independent in both M_1 and M_2.
- If $I + v_1 \notin \mathcal{I}_2$, $\exists C_2(I, v_1)$ a circuit in M_2, and choosing $v_2 \in C_2(I, v_1)$ s.t. $v_2 \neq v_1$ leads to $I + v_1 - v_2$ which (because $\text{span}_2(I) = \text{span}(I + v_1 - v_2)$) is again independent in M_2. It is also independent in M_1.
- Next choose a $v_3 \in \text{span}_1(I) - \text{span}_1(I - v_2)$ to recover what was lost in $I \cup \{v_1\}$ when we removed v_2 from it.
- Then $\text{span}_1(I) = \text{span}_1(I - v_2 + v_3)$.
- Moreover, since $I + v_1 \in \mathcal{I}_1$, $v_1 \notin \text{span}_1(I)$, so $\text{span}_1(I + v_1) = \text{span}_1(I + v_1 - v_2 + v_3)$.

Matroid Intersection Algorithm Idea

- Consider two matroids $M_1 = (V, \mathcal{I}_1)$ and $M_2 = (V, \mathcal{I}_2)$ and start with any $I \in \mathcal{I}_1 \cap \mathcal{I}_2$.
- Consider some $v_1 \notin \text{span}_1(I)$, so that $I + v_1 \in \mathcal{I}_1$.
- If $I + v_1 \in \mathcal{I}_2$, then v_1 is "augmenting", and we can augment I to $I + v_1$ and still be independent in both M_1 and M_2.
- If $I + v_1 \notin \mathcal{I}_2$, $\exists C_2(I, v_1)$ a circuit in M_2, and choosing $v_2 \in C_2(I, v_1)$ s.t. $v_2 \neq v_1$ leads to $I + v_1 - v_2$ which (because $\text{span}_2(I) = \text{span}(I + v_1 - v_2)$) is again independent in M_2. It is also independent in M_1.
- Next choose a $v_3 \in \text{span}_1(I) - \text{span}_1(I - v_2)$ to recover what was lost in $I \cup \{v_1\}$ when we removed v_2 from it.
- Then $\text{span}_1(I) = \text{span}_1(I - v_2 + v_3)$.
- Moreover, since $I + v_1 \in \mathcal{I}_1$, $v_1 \notin \text{span}_1(I)$, so $\text{span}_1(I + v_1) = \text{span}_1(I + v_1 - v_2 + v_3)$.
- But $I + v_1 - v_2 + v_3$ might not be independent in M_2 again, so we need to find an $v_4 \in C_2(I + v_1 - v_2, v_3)$ to remove, and so on.

Matroid Intersection Algorithm Idea

- Hopefully (eventually) we'll find an odd length sequence $S = (v_1, v_2, \ldots, v_n)$ such that we will be independent in both M_1 andM_2 and thus be one greater in size than I.

Matroid Intersection Algorithm Idea

- Hopefully (eventually) we'll find an odd length sequence $S = (v_1, v_2, \ldots, v_n)$ such that we will be independent in both M_1 andM_2 and thus be one greater in size than I.
- We then replace I with $I \ominus S$ (quite analogous to the bipartite matching case), and start again.

Alternating and Augmenting Sequences

- Let I be an intersection of two matroids $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ (i.e., $I \in \mathcal{I}_1 \cap \mathcal{I}_2$).

Alternating and Augmenting Sequences

- Let I be an intersection of two matroids $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ (i.e., $I \in \mathcal{I}_1 \cap \mathcal{I}_2$).
- Let $S = (e_1, e_2, \dots, e_s)$ be a sequence of distinct elements, where $e_i \in E - I$ for i odd, and $e_i \in I$ for i even, and let $S_i = (e_1, e_2, \dots, e_i)$. We say that S is an alternating sequence w.r.t. I if the following are true.

Alternating and Augmenting Sequences

- Let I be an intersection of two matroids $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ (i.e., $I \in \mathcal{I}_1 \cap \mathcal{I}_2$).
- Let $S = (e_1, e_2, \ldots, e_s)$ be a sequence of distinct elements, where $e_i \in E - I$ for i odd, and $e_i \in I$ for i even, and let $S_i = (e_1, e_2, \ldots, e_i)$. We say that S is an alternating sequence w.r.t. I if the following are true.
  1. $I + e_1 \in \mathcal{I}_1$

Alternating and Augmenting Sequences

- Let I be an intersection of two matroids $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ (i.e., $I \in \mathcal{I}_1 \cap \mathcal{I}_2$).
- Let $S = (e_1, e_2, \dots, e_s)$ be a sequence of distinct elements, where $e_i \in E - I$ for i odd, and $e_i \in I$ for i even, and let $S_i = (e_1, e_2, \dots, e_i)$. We say that S is an alternating sequence w.r.t. I if the following are true.
  1. $I + e_1 \in \mathcal{I}_1$
  2. For all even i, $\text{span}_2(I \ominus S_i) = \text{span}_2(I)$ which implies that $I \ominus S_i \in \mathcal{I}_2$.

Alternating and Augmenting Sequences

- Let I be an intersection of two matroids $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ (i.e., $I \in \mathcal{I}_1 \cap \mathcal{I}_2$).
- Let $S = (e_1, e_2, \dots, e_s)$ be a sequence of distinct elements, where $e_i \in E - I$ for i odd, and $e_i \in I$ for i even, and let $S_i = (e_1, e_2, \dots, e_i)$. We say that S is an alternating sequence w.r.t. I if the following are true.
  1. $I + e_1 \in \mathcal{I}_1$
  2. For all even i, $\text{span}_2(I \ominus S_i) = \text{span}_2(I)$ which implies that $I \ominus S_i \in \mathcal{I}_2$.
  3. For all odd i, $\text{span}_1(S \ominus S_i) = \text{span}_1(I + e_1)$, and therefore $I \ominus S_i \in \mathcal{I}_1$.

Alternating and Augmenting Sequences

- Let I be an intersection of two matroids $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ (i.e., $I \in \mathcal{I}_1 \cap \mathcal{I}_2$).
- Let $S = (e_1, e_2, \dots, e_s)$ be a sequence of distinct elements, where $e_i \in E - I$ for i odd, and $e_i \in I$ for i even, and let $S_i = (e_1, e_2, \dots, e_i)$. We say that S is an alternating sequence w.r.t. I if the following are true.
  1. $I + e_1 \in \mathcal{I}_1$
  2. For all even i, $\text{span}_2(I \ominus S_i) = \text{span}_2(I)$ which implies that $I \ominus S_i \in \mathcal{I}_2$.
  3. For all odd i, $\text{span}_1(S \ominus S_i) = \text{span}_1(I + e_1)$, and therefore $I \ominus S_i \in \mathcal{I}_1$.
- Lastly, if also, $|S| = s$ is odd, and $I \ominus S \in \mathcal{I}_2$, then S is called an augmenting sequence w.r.t. I.

Alternating and Augmenting Sequences

- If I admits an augmenting sequence S, then the above argument shows that $I \ominus S$ is independent in M_1, independent in M_2, and also we have that $|I| + 1 = |I \ominus S|$.

Alternating and Augmenting Sequences

- If I admits an augmenting sequence S, then the above argument shows that $I \ominus S$ is independent in M_1, independent in M_2, and also we have that $|I| + 1 = |I \ominus S|$.
- Thus, by finding augmenting sequences, we can increase the size of the matroid intersection until we stop. Moreover, if there is an augmenting sequence, then the intersection is not maximum.

Alternating and Augmenting Sequences

- If I admits an augmenting sequence S, then the above argument shows that $I \ominus S$ is independent in M_1, independent in M_2, and also we have that $|I| + 1 = |I \ominus S|$.
- Thus, by finding augmenting sequences, we can increase the size of the matroid intersection until we stop. Moreover, if there is an augmenting sequence, then the intersection is not maximum.
- We next wish to show that, if the intersection is maximum, then there is an augmenting sequence.

Matroid Intersection Algorithm

- Given two matroids $M_1 = (V, \mathcal{I}_1)$ and $M_2 = (V, \mathcal{I}_2)$

Matroid Intersection Algorithm

- Given two matroids $M_1 = (V, \mathcal{I}_1)$ and $M_2 = (V, \mathcal{I}_2)$
- Our goal is to find the maximum cardinality set I such that $I \in \mathcal{I}_1$ and $I \in \mathcal{I}_2$.

Matroid Intersection Algorithm

- Given two matroids $M_1 = (V, \mathcal{I}_1)$ and $M_2 = (V, \mathcal{I}_2)$
- Our goal is to find the maximum cardinality set I such that $I \in \mathcal{I}_1$ and $I \in \mathcal{I}_2$.
- The algorithm described becomes:

Algorithm 9.1: Alternating Path Matroid Intersection

```
Let I be an arbitrary (including empty) independent set in two matroids M_1 and M_2 ;
while There exists an augmenting sequence S do
    I ← I ⊖ S ;
```

Matroid Intersection Algorithm

- Given two matroids $M_1 = (V, \mathcal{I}_1)$ and $M_2 = (V, \mathcal{I}_2)$
- Our goal is to find the maximum cardinality set I such that $I \in \mathcal{I}_1$ and $I \in \mathcal{I}_2$.
- The algorithm described becomes:

Algorithm 9.1: Alternating Path Matroid Intersection

```
Let I be an arbitrary (including empty) independent set in two matroids M_1 and M_2 ;
while There exists an augmenting sequence S do
    I ← I ⊖ S ;
```

- This can be made to run in $O(m^2 R + mR^2 c(m))$ where $m = |V|$, $R =$ max rank of the two matroids, and $c(m)$ is the max independence testing cost for the two matroids, but faster algorithms exist as well (see Schrijver-2003).

Border graphs

- We construct an auxiliary directed bipartite graph (border graph) $B(I) = (E \setminus I, I, Z)$, relative to the current I, that will help us with this problem. The graph has only directed edges from $E \setminus I$ to I, or from I back to $E \setminus I$.

Border graphs

- We construct an auxiliary directed bipartite graph (border graph) $B(I) = (E \setminus I, I, Z)$, relative to the current I, that will help us with this problem. The graph has only directed edges from $E \setminus I$ to I, or from I back to $E \setminus I$.
- Left-going edges: For each $e_i \in \text{span}_1(I) \setminus I$, create $\leftarrow$ edge $(e_j, e_i) \in Z$ for any $e_j \in C_1(I, e_i) \setminus \{e_i\}$.

Border graphs

- We construct an auxiliary directed bipartite graph (border graph) $B(I) = (E \setminus I, I, Z)$, relative to the current I, that will help us with this problem. The graph has only directed edges from $E \setminus I$ to I, or from I back to $E \setminus I$.
- Left-going edges: For each $e_i \in \text{span}_1(I) \setminus I$, create $\leftarrow$ edge $(e_j, e_i) \in Z$ for any $e_j \in C_1(I, e_i) \setminus \{e_i\}$.
- If $e_i \notin \text{span}_1(I)$, then e_i has in-degree zero (a source).

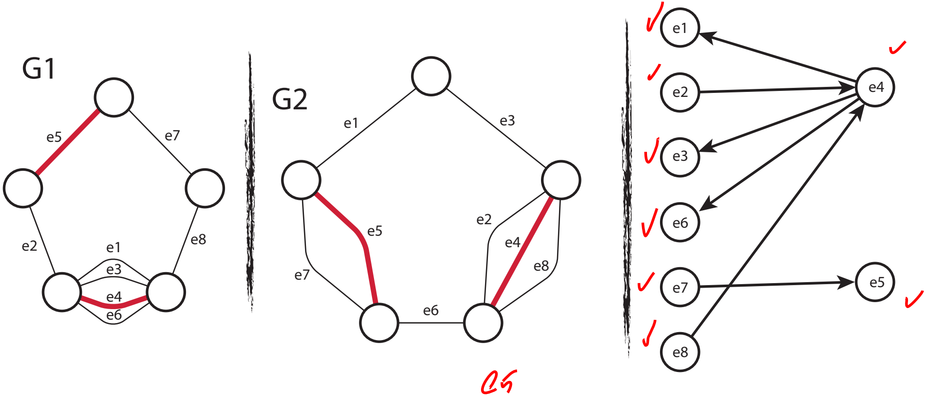
Border graphs

- We construct an auxiliary directed bipartite graph (border graph) $B(I) = (E \setminus I, I, Z)$, relative to the current I, that will help us with this problem. The graph has only directed edges from $E \setminus I$ to I, or from I back to $E \setminus I$.
- Left-going edges: For each $e_i \in \text{span}_1(I) \setminus I$, create $\leftarrow$ edge $(e_j, e_i) \in Z$ for any $e_j \in C_1(I, e_i) \setminus \{e_i\}$.
- If $e_i \notin \text{span}_1(I)$, then e_i has in-degree zero (a source).
- Right-going edges: For each $e_i \in \text{span}_2(I) \setminus I$, create $\rightarrow$ edge $(e_i, e_j) \in Z$ for any $e_j \in C_2(I, e_i) \setminus \{e_i\}$.

Border graphs

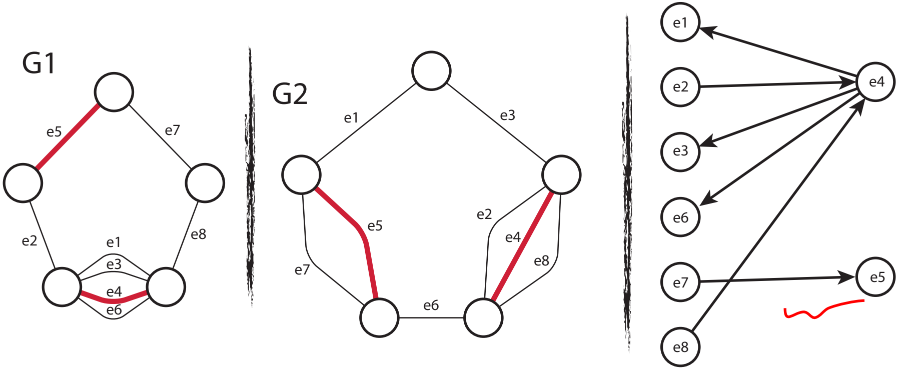
- We construct an auxiliary directed bipartite graph (border graph) $B(I) = (E \setminus I, I, Z)$, relative to the current I, that will help us with this problem. The graph has only directed edges from $E \setminus I$ to I, or from I back to $E \setminus I$.
- Left-going edges: For each $e_i \in \text{span}_1(I) \setminus I$, create $\leftarrow$ edge $(e_j, e_i) \in Z$ for any $e_j \in C_1(I, e_i) \setminus \{e_i\}$.
- If $e_i \notin \text{span}_1(I)$, then e_i has in-degree zero (a source).
- Right-going edges: For each $e_i \in \text{span}_2(I) \setminus I$, create $\rightarrow$ edge $(e_i, e_j) \in Z$ for any $e_j \in C_2(I, e_i) \setminus \{e_i\}$.
- If $e_i \notin \text{span}_2(I)$, then e_i has out-degree zero (a sink).

Border graph Example



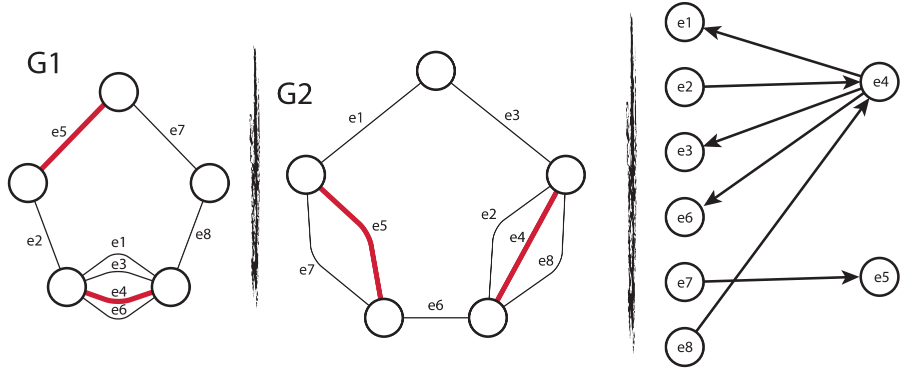
- $\{e_2, e_7, e_8\}$ are sources and $\{e_1, e_3, e_6\}$ are sinks.

Border graph Example



- $\{e_2, e_7, e_8\}$ are sources and $\{e_1, e_3, e_6\}$ are sinks.
- Augmenting sequences are (e_2, e_4, e_1), (e_2, e_4, e_3), and (e_2, e_4, e_6), all of which are dipaths in the Border graph.

Border graph Example



- $\{e_2, e_7, e_8\}$ are sources and $\{e_1, e_3, e_6\}$ are sinks.
- Augmenting sequences are (e_2, e_4, e_1), (e_2, e_4, e_3), and (e_2, e_4, e_6), all of which are dipaths in the Border graph.
- Are there others? yes.

Identifying Augmenting Sequences

Lemma 3.1

If S is a source-sink path in $B(I)$, and there is no shorter source-sink path between the same source and sink (i.e., there are no short-cuts), then S is an augmenting sequence w.r.t. I.

Identifying Augmenting Sequences

Lemma 3.1

If S is a source-sink path in $B(I)$, and there is no shorter source-sink path between the same source and sink (i.e., there are no short-cuts), then S is an augmenting sequence w.r.t. I.

Lemma 3.2

Let I and J be intersections such that $|I| + 1 = |J|$. Then there exists a source-sink path S in $B(I)$ where $S \subseteq I \ominus J$.

Identifying Augmenting Sequences

Theorem 3.3

Let I_p and I_{p+1} be intersections of M_1 and M_2 with p and $p+1$ elements respectively. Then there exists an augmenting sequence $S \subseteq I_p \ominus I_{p+1}$ w.r.t. I_p.

Identifying Augmenting Sequences

Theorem 3.3

Let I_p and I_{p+1} be intersections of M_1 and M_2 with p and $p+1$ elements respectively. Then there exists an augmenting sequence $S \subseteq I_p \ominus I_{p+1}$ w.r.t. I_p.

Theorem 3.4

An intersection is of maximum cardinality iff it admits no augmenting sequence.

Identifying Augmenting Sequences

Theorem 3.3

Let I_p and I_{p+1} be intersections of M_1 and M_2 with p and $p+1$ elements respectively. Then there exists an augmenting sequence $S \subseteq I_p \ominus I_{p+1}$ w.r.t. I_p.

Theorem 3.4

An intersection is of maximum cardinality iff it admits no augmenting sequence.

Theorem 3.5

For any intersection I, there exists a maximum cardinality intersection I^* such that $\text{span}_1(I) \subseteq \text{span}_1(I^*)$ and $\text{span}_2(I) \subseteq \text{span}_2(I^*)$.

Matroid Partition Problem

- Suppose $M_i = (E, \mathcal{I}_i)$ is a matroid and that we have k of them on the same ground set E.

Matroid Partition Problem

- Suppose $M_i = (E, \mathcal{I}_i)$ is a matroid and that we have k of them on the same ground set E.
- We wish to, if possible, partition E into k blocks, $I_1, i \in \{1, 2, \ldots, k\}$ where $I_i \in \mathcal{I}_i$.

Matroid Partition Problem

- Suppose $M_i = (E, \mathcal{I}_i)$ is a matroid and that we have k of them on the same ground set E.
- We wish to, if possible, partition E into k blocks, $I_1, i \in \{1, 2, \dots, k\}$ where $I_i \in \mathcal{I}_i$.
- Moreover, we want partition to be lexicographically maximum, that is $|I_1|$ is maximum, $|I_2|$ is maximum given $|I_1|$, and so on.

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \dots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{5}$$

where r_i is the rank function of M_i.

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \dots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{5}$$

where r_i is the rank function of M_i.

- Now, if all matroids are the same $M_i = M$ for all i, we get condition

$$|A| \leq kr(A) \;\; \forall A \subseteq E \tag{6}$$

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \dots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{5}$$

where r_i is the rank function of M_i.

- Now, if all matroids are the same $M_i = M$ for all i, we get condition
$$|A| \leq kr(A) \;\; \forall A \subseteq E \tag{6}$$
- But considering vector of all ones $\mathbf{1} \in \mathbb{R}_+^E$, this is the same as
$$\frac{1}{k}\mathbf{1}(A) \leq r(A) \;\; \forall A \subseteq E \tag{7}$$

Matroid Partition Problem and Submodular Function Minimization

- Recall definition of matroid polytope

$$P_r = \left\{ y \in \mathbb{R}_+^E : y(A) \leq r(A) \text{ for all } A \subseteq E \right\} \tag{11}$$

Matroid Partition Problem and Submodular Function Minimization

- Recall definition of matroid polytope

$$P_r = \left\{ y \in \mathbb{R}_+^E : y(A) \le r(A) \text{ for all } A \subseteq E \right\} \tag{11}$$

- Then we see that this special case of the matroid partition problem is just testing if $\frac{1}{k}\mathbf{1} \in P_r$, a problem of testing the membership in matroid polyhedra.

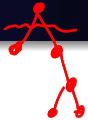
Matroid Partition Problem and Submodular Function Minimization

- Recall definition of matroid polytope

$$P_r = \left\{ y \in \mathbb{R}_+^E : y(A) \leq r(A) \text{ for all } A \subseteq E \right\} \tag{11}$$

- Then we see that this special case of the matroid partition problem is just testing if $\frac{1}{k}\mathbf{1} \in P_r$, a problem of testing the membership in matroid polyhedra.
- We also see that this is essentially a special case of submodular function minimization, namely finding A that minimizes $r(A) - \frac{1}{k}\mathbf{1}(A)$.

Matroid Partition Problem and Submodular Function Minimization



- Recall definition of matroid polytope

$$P_r = \left\{ y \in \mathbb{R}_+^E : y(A) \leq r(A) \text{ for all } A \subseteq E \right\} \tag{11}$$

- Then we see that this special case of the matroid partition problem is just testing if $\frac{1}{k}\mathbf{1} \in P_r$, a problem of testing the membership in matroid polyhedra.
- We also see that this is essentially a special case of submodular function minimization, namely finding A that minimizes $r(A) - \frac{1}{k}\mathbf{1}(A)$.
- In the general case, we are looking for an A that minimizes $\sum_i r_i(A) - \frac{1}{k}\mathbf{1}(A)$, and a sum of submodular functions is submodular (in fact, a sum of matroid rank functions is a type of polymatroid rank function Exercise).

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \ldots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{5}$$

where r_i is the rank function of M_i.

Proof.

- Suppose I is partitionable into subsets $I_i, i = 1, \ldots, k$.

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \dots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{5}$$

where r_i is the rank function of M_i.

Proof.

- Suppose I is partitionable into subsets $I_i, i = 1, \dots, k$.
- Since it is a partition, for any $A \subseteq I$, we must have

$$|A| = \sum_{i=1}^{k} |I_i \cap A| \tag{8}$$

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \dots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{5}$$

where r_i is the rank function of M_i.

Proof.

- Suppose I is partitionable into subsets $I_i, i = 1, \dots, k$.
- Since it is a partition, for any $A \subseteq I$, we must have

$$|A| = \sum_{i=1}^{k} |I_i \cap A| \tag{8}$$

- and moreover for each independent set I_i, we have

$$|I_i \cap A| \leq r_i(A) \tag{9}$$

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \dots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{5}$$

where r_i is the rank function of M_i.

Proof.

- which immediately gives:

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{10}$$

  proving the first half of the theorem.

□

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \dots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \quad (5)$$

where r_i is the rank function of M_i.

- To prove the converse, we start by assuming that Eq. 5 is true.

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \dots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{5}$$

where r_i is the rank function of M_i.

- To prove the converse, we start by assuming that Eq. 5 is true.
- We derive an algorithm that, under assumption Eq. 5, will produce such a set of independent sets, and if Eq. 5 is false, just halts.

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \dots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{5}$$

where r_i is the rank function of M_i.

- To prove the converse, we start by assuming that Eq. 5 is true.
- We derive an algorithm that, under assumption Eq. 5, will produce such a set of independent sets, and if Eq. 5 is false, just halts.
- The algorithm needn't verify the assumption, rather it runs, and if halts it ensures Eq. 5 is false. If it succeeds, Eq. 5 is true.

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \dots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{5}$$

where r_i is the rank function of M_i.

- To prove the converse, we start by assuming that Eq. 5 is true.
- We derive an algorithm that, under assumption Eq. 5, will produce such a set of independent sets, and if Eq. 5 is false, just halts.
- The algorithm needn't verify the assumption, rather it runs, and if halts it ensures Eq. 5 is false. If it succeeds, Eq. 5 is true.
- We give algorithm for $I = E$, but any $I \subseteq E$ can be used instead.

Matroid Partition Algorithm (and proof)

1. The algorithm starts with a set of k empty sets, $I_k = \emptyset, i = 1, 2, \ldots, k$. J is our index set, so $J = \{1, 2, \ldots, k\}$.

Matroid Partition Algorithm (and proof)

1. The algorithm starts with a set of k empty sets, $I_k = \emptyset, i = 1, 2, \ldots, k$. J is our index set, so $J = \{1, 2, \ldots, k\}$.
2. We are going to create a sequence of subsets $(S_0, S_1, \ldots)$, starting with $S_0 = E$, and the others are defined by the algorithm below.

For each S_i, there will be an associated $I_{j(i)}$. For simplicity, we will be calling/renaming $I_{j(i)} \triangleq I_i$ w.l.o.g.

there might be duplicates, i.e., could have $j(i) = j(i')$ for $i \neq i'$.

Matroid Partition Algorithm (and proof)

1. The algorithm starts with a set of k empty sets, $I_k = \emptyset, i = 1, 2, \ldots, k$. J is our index set, so $J = \{1, 2, \ldots, k\}$.
2. We are going to create a sequence of subsets $(S_0, S_1, \ldots)$, starting with $S_0 = E$, and the others are defined by the algorithm below.
3. Assume there is an $e \in E$ s.t. $e \notin \bigcup_i I_i$ (otherwise we're done).

Matroid Partition Algorithm (and proof)

1. The algorithm starts with a set of k empty sets, $I_k = \emptyset, i = 1, 2, \dots, k$. J is our index set, so $J = \{1, 2, \dots, k\}$.
2. We are going to create a sequence of subsets $(S_0, S_1, \dots)$, starting with $S_0 = E$, and the others are defined by the algorithm below.
3. Assume there is an $e \in E$ s.t. $e \notin \bigcup_i I_i$ (otherwise we're done).
4. If $\forall i$, $|I_i| \geq r_i(E)$, then $|E| \geq |\{e\} + \bigcup_i I_i| > \sum_i r_i(E)$ which is not possible by assumption.

$|E| > \sum_i r_i(E)$

Matroid Partition Algorithm (and proof)

1. The algorithm starts with a set of k empty sets, $I_k = \emptyset, i = 1, 2, \ldots, k$. J is our index set, so $J = \{1, 2, \ldots, k\}$.
2. We are going to create a sequence of subsets $(S_0, S_1, \ldots)$, starting with $S_0 = E$, and the others are defined by the algorithm below.
3. Assume there is an $e \in E$ s.t. $e \notin \bigcup_i I_i$ (otherwise we're done).
4. If $\forall i$, $|I_i| \geq r_i(E)$, then $|E| \geq |\{e\} + \bigcup_i I_i| > \sum_i r_i(E)$ which is not possible by assumption.
5. Therefore, $\exists$ smallest i s.t. $|I_i| < r_i(E)$. W.l.o.g. name $i = 1$, and let $S_1 \stackrel{\text{def}}{=} \text{span}_1(I_1) = S_0 \cap \text{span}_1(I_1 \cap S_0) = E \cap \text{span}_1(I_1 \cap E)$.

strict.

Matroid Partition Algorithm (and proof)

1. The algorithm starts with a set of k empty sets, $I_k = \emptyset, i = 1, 2, \ldots, k$. J is our index set, so $J = \{1, 2, \ldots, k\}$.
2. We are going to create a sequence of subsets $(S_0, S_1, \ldots)$, starting with $S_0 = E$, and the others are defined by the algorithm below.
3. Assume there is an $e \in E$ s.t. $e \notin \bigcup_i I_i$ (otherwise we're done).
4. If $\forall i$, $|I_i| \geq r_i(E)$, then $|E| \geq |\{e\} + \bigcup_i I_i| > \sum_i r_i(E)$ which is not possible by assumption.
5. Therefore, $\exists$ smallest i s.t. $|I_i| < r_i(E)$. W.l.o.g. name $i = 1$, and let $S_1 \stackrel{\text{def}}{=} \text{span}_1(I_1) = S_0 \cap \text{span}_1(I_1 \cap S_0) = E \cap \text{span}_1(I_1 \cap E)$.
6. Assume $e \in S_1$. Now, similarly, if $\forall i$, $|I_i \cap S_1| \geq r_i(S_1)$, then $|S_1| \geq |\{e\} + \bigcup_i (I_i \cap S_1)| > \sum_i r_i(S_1)$ which is again not possible by assumption (recall $\forall i, e \notin I_i$).

$(\{e\} + (S_1 \cap \bigcup_i I_i)) \Rightarrow |S_1| > \sum_i r_i(S_1)$

Matroid Partition Algorithm (and proof)

1. The algorithm starts with a set of k empty sets, $I_k = \emptyset, i = 1, 2, \ldots, k$. J is our index set, so $J = \{1, 2, \ldots, k\}$.
2. We are going to create a sequence of subsets $(S_0, S_1, \ldots)$, starting with $S_0 = E$, and the others are defined by the algorithm below.
3. Assume there is an $e \in E$ s.t. $e \notin \bigcup_i I_i$ (otherwise we're done).
4. If $\forall i$, $|I_i| \geq r_i(E)$, then $|E| \geq |\{e\} + \bigcup_i I_i| > \sum_i r_i(E)$ which is not possible by assumption.
5. Therefore, $\exists$ smallest i s.t. $|I_i| < r_i(E)$. W.l.o.g. name $i = 1$, and let $S_1 \stackrel{\text{def}}{=} \text{span}_1(I_1) = S_0 \cap \text{span}_1(I_1 \cap S_0) = E \cap \text{span}_1(I_1 \cap E)$.
6. Assume $e \in S_1$. Now, similarly, if $\forall i$, $|I_i \cap S_1| \geq r_i(S_1)$, then $|S_1| \geq |\{e\} + \bigcup_i (I_i \cap S_1)| > \sum_i r_i(S_1)$ which is again not possible by assumption (recall $\forall i, e \notin I_i$).
7. Therefore, $\exists$ smallest i' s.t. $|I_{i'} \cap S_1| < r_{i'}(S_1)$. W.l.o.g. name $i' = 2$, and let $S_2 \stackrel{\text{def}}{=} S_1 \cap \text{span}_2(I_2 \cap S_1)$. $S_2 \subseteq S_1$

Matroid Partition Algorithm (and proof)

8. Note $S_2 \subseteq S_1$. Moreover, we have $S_2 \subset S_1$ (proper) since

$$\begin{aligned} r_2(S_2) &= r_2(S_1 \cap \text{span}_2(I_2 \cap S_1)) && (12)\\ &\leq r_2(\text{span}_2(I_2 \cap S_1)) && (13)\\ &= r_2(I_2 \cap S_1) && (14)\\ &\leq |I_2 \cap S_1| < r_2(S_1) && (15) \end{aligned}$$

and if the rank decreases, the sets can't be equal.

Matroid Partition Algorithm (and proof)

8. Note $S_2 \subseteq S_1$. Moreover, we have $S_2 \subset S_1$ (proper) since

$$r_2(S_2) = r_2(S_1 \cap \text{span}_2(I_2 \cap S_1)) \quad (12)$$
$$\leq r_2(\text{span}_2(I_2 \cap S_1)) \quad (13)$$
$$= r_2(I_2 \cap S_1) \quad (14)$$
$$\leq |I_2 \cap S_1| < r_2(S_1) \quad (15)$$

and if the rank decreases, the sets can't be equal.

9. Iterating on j, assume $e \in S_j$. Now, similarly, if $\forall i$, $|I_i \cap S_j| \geq r_i(S_j)$, then $|S_j| \geq |\{e\} + \bigcup_i (I_i \cap S_j)| > \sum_i r_i(S_j)$ which is again not possible by assumption (recall $\forall i$, $e \notin I_i$). $\Rightarrow |S_j| > \sum_i r_i(S_j)$

Matroid Partition Algorithm (and proof)

8. Note $S_2 \subseteq S_1$. Moreover, we have $S_2 \subset S_1$ (proper) since
$$\begin{aligned} r_2(S_2) &= r_2(S_1 \cap \text{span}_2(I_2 \cap S_1)) && (12)\\ &\leq r_2(\text{span}_2(I_2 \cap S_1)) && (13)\\ &= r_2(I_2 \cap S_1) && (14)\\ &\leq |I_2 \cap S_1| < r_2(S_1) && (15)\end{aligned}$$
and if the rank decreases, the sets can't be equal.

9. Iterating on j, assume $e \in S_j$. Now, similarly, if $\forall i$, $|I_i \cap S_j| \geq r_i(S_j)$, then $|S_j| \geq |\{e\} + \bigcup_i (I_i \cap S_j)| > \sum_i r_i(S_j)$ which is again not possible by assumption (recall $\forall i, e \notin I_i$).

10. Therefore, $\exists$ smallest i' s.t. $|I_{i'} \cap S_j| < r_i(S_j)$. W.l.o.g. name $i' = j + 1$, and let $S_{j+1} \stackrel{\text{def}}{=} S_j \cap \text{span}_{j+1}(I_{j+1} \cap S_j)$.

Matroid Partition Algorithm (and proof)

8. Note $S_2 \subseteq S_1$. Moreover, we have $S_2 \subset S_1$ (proper) since

$$r_2(S_2) = r_2(S_1 \cap \text{span}_2(I_2 \cap S_1)) \tag{12}$$
$$\leq r_2(\text{span}_2(I_2 \cap S_1)) \tag{13}$$
$$= r_2(I_2 \cap S_1) \tag{14}$$
$$\leq |I_2 \cap S_1| < r_2(S_1) \tag{15}$$

   and if the rank decreases, the sets can't be equal.

9. Iterating on j, assume $e \in S_j$. Now, similarly, if $\forall i$, $|I_i \cap S_j| \geq r_i(S_j)$, then $|S_j| \geq |\{e\} + \bigcup_i (I_i \cap S_j)| > \sum_i r_i(S_j)$ which is again not possible by assumption (recall $\forall i, e \notin I_i$).

10. Therefore, $\exists$ smallest i' s.t. $|I_{i'} \cap S_j| < r_i(S_j)$. W.l.o.g. name $i' = j + 1$, and let $S_{j+1} \stackrel{\text{def}}{=} S_j \cap \text{span}_{j+1}(I_{j+1} \cap S_j)$.

11. And we have $S_{j+1} \subset S_j$. by same argument as equations 12–15.

Matroid Partition Algorithm (and proof)

8. Note $S_2 \subseteq S_1$. Moreover, we have $S_2 \subset S_1$ (proper) since
$$r_2(S_2) = r_2(S_1 \cap \text{span}_2(I_2 \cap S_1)) \tag{12}$$
$$\leq r_2(\text{span}_2(I_2 \cap S_1)) \tag{13}$$
$$= r_2(I_2 \cap S_1) \tag{14}$$
$$\leq |I_2 \cap S_1| < r_2(S_1) \tag{15}$$
and if the rank decreases, the sets can't be equal.

9. Iterating on j, assume $e \in S_j$. Now, similarly, if $\forall i$, $|I_i \cap S_j| \geq r_i(S_j)$, then $|S_j| \geq |\{e\} + \bigcup_i (I_i \cap S_j)| > \sum_i r_i(S_j)$ which is again not possible by assumption (recall $\forall i, e \notin I_i$).

10. Therefore, $\exists$ smallest i' s.t. $|I_{i'} \cap S_j| < r_i(S_j)$. W.l.o.g. name $i' = j + 1$, and let $S_{j+1} \stackrel{\text{def}}{=} S_j \cap \text{span}_{j+1}(I_{j+1} \cap S_j)$.

11. And we have $S_{j+1} \subset S_j$.

12. In general, we have $S_0 \supset S_1 \supset S_2 \supset \cdots \supset S_{h-1}$ as long as $e \in S_j$ for $j = 0, \ldots, h-1$. Note that the I_i's might be being reused.

so might have $h > k$

Matroid Partition Algorithm (and proof)

13. Due to the strict monotone decreasing, at some point we must get to an h such that $e \notin S_h$. Two things can then happen:

(could be that $S_h = \phi$).

Matroid Partition Algorithm (and proof)

13. Due to the strict monotone decreasing, at some point we must get to an h such that $e \notin S_h$. Two things can then happen:
14. If $I_h + e \in \mathcal{I}_h$ then set $I_h \leftarrow I_h + e$, thereby growing our set of independent sets. Empty all S_i's, and GOTO line 3, and continue.

Matroid Partition Algorithm (and proof)

13. Due to the strict monotone decreasing, at some point we must get to an h such that $e \notin S_h$. Two things can then happen:
14. If $I_h + e \in \mathcal{I}_h$ then set $I_h \leftarrow I_h + e$, thereby growing our set of independent sets. Empty all S_i's, and GOTO line 3, and continue.
15. If, on the other hand, $I_h + e \notin M_h$ then there is a (nec. unique) circuit $C_h(I_h, e)$ created in M_h when adding e to I_h.

Matroid Partition Algorithm (and proof)

13. Due to the strict monotone decreasing, at some point we must get to an h such that $e \notin S_h$. Two things can then happen:
14. If $I_h + e \in \mathcal{I}_h$ then set $I_h \leftarrow I_h + e$, thereby growing our set of independent sets. Empty all S_i's, and GOTO line 3, and continue.
15. If, on the other hand, $I_h + e \notin M_h$ then there is a (nec. unique) circuit $C_h(I_h, e)$ created in M_h when adding e to I_h.
16. Now, suppose $C_h(I_h, e) \subseteq S_{h-1}$. Then since $e \in \text{span}_h(I_h)$, this gives $C_h(I_h, e) \subseteq \text{span}_h(I_h \cap S_{h-1})$. And since $S_h \stackrel{\text{def}}{=} S_{h-1} \cap \text{span}_h(I_h \cap S_{h-1})$, this also implies $C_h(I_h, e) \subseteq S_h$, which is impossible since $e \notin S_h$ (from line 13 or 17).

Matroid Partition Algorithm (and proof)

13. Due to the strict monotone decreasing, at some point we must get to an h such that $e \notin S_h$. Two things can then happen:
14. If $I_h + e \in \mathcal{I}_h$ then set $I_h \leftarrow I_h + e$, thereby growing our set of independent sets. Empty all S_i's, and GOTO line 3, and continue.
15. If, on the other hand, $I_h + e \notin M_h$ then there is a (nec. unique) circuit $C_h(I_h, e)$ created in M_h when adding e to I_h.
16. Now, suppose $C_h(I_h, e) \subseteq S_{h-1}$. Then since $e \in \text{span}_h(I_h)$, this gives $C_h(I_h, e) \subseteq \text{span}_h(I_h \cap S_{h-1})$. And since $S_h \stackrel{\text{def}}{=} S_{h-1} \cap \text{span}_h(I_h \cap S_{h-1})$, this also implies $C_h(I_h, e) \subseteq S_h$, which is impossible since $e \notin S_h$ (from line 13 or 17).
17. Thus, we chose an element $e' \in C_h(I_h, e) \setminus S_{h-1}$. Note that $e' \neq e$ because $e \in S_j$ for all $j < h$.

Matroid Partition Algorithm (and proof)

13. Due to the strict monotone decreasing, at some point we must get to an h such that $e \notin S_h$. Two things can then happen:
14. If $I_h + e \in \mathcal{I}_h$ then set $I_h \leftarrow I_h + e$, thereby growing our set of independent sets. Empty all S_i's, and GOTO line 3, and continue.
15. If, on the other hand, $I_h + e \notin M_h$ then there is a (nec. unique) circuit $C_h(I_h, e)$ created in M_h when adding e to I_h.
16. Now, suppose $C_h(I_h, e) \subseteq S_{h-1}$. Then since $e \in \text{span}_h(I_h)$, this gives $C_h(I_h, e) \subseteq \text{span}_h(I_h \cap S_{h-1})$. And since $S_h \stackrel{\text{def}}{=} S_{h-1} \cap \text{span}_h(I_h \cap S_{h-1})$, this also implies $C_h(I_h, e) \subseteq S_h$, which is impossible since $e \notin S_h$ (from line 13 or 17).
17. Thus, we chose an element $e' \in C_h(I_h, e) \setminus S_{h-1}$. Note that $e' \neq e$ because $e \in S_j$ for all $j < h$.
18. Then, we update $I_h \leftarrow I_h + e - e'$ which retains I_h's independence.

Matroid Partition Algorithm (and proof)

13. Due to the strict monotone decreasing, at some point we must get to an h such that $e \notin S_h$. Two things can then happen:
14. If $I_h + e \in \mathcal{I}_h$ then set $I_h \leftarrow I_h + e$, thereby growing our set of independent sets. Empty all S_i's, and GOTO line 3, and continue.
15. If, on the other hand, $I_h + e \notin M_h$ then there is a (nec. unique) circuit $C_h(I_h, e)$ created in M_h when adding e to I_h.
16. Now, suppose $C_h(I_h, e) \subseteq S_{h-1}$. Then since $e \in \text{span}_h(I_h)$, this gives $C_h(I_h, e) \subseteq \text{span}_h(I_h \cap S_{h-1})$. And since $S_h \stackrel{\text{def}}{=} S_{h-1} \cap \text{span}_h(I_h \cap S_{h-1})$, this also implies $C_h(I_h, e) \subseteq S_h$, which is impossible since $e \notin S_h$ (from line 13 or 17).
17. Thus, we chose an element $e' \in C_h(I_h, e) \setminus S_{h-1}$. Note that $e' \neq e$ because $e \in S_j$ for all $j < h$.
18. Then, we update $I_h \leftarrow I_h + e - e'$ which retains I_h's independence.
19. Next, decrement $h \leftarrow h - 1$, update $e \leftarrow e'$, and GOTO line 14.

Matroid Partition Algorithm (and proof)

20. Last, we must make sure that the loop from lines 14 through 19 terminate correctly, as h is decremented down.

Matroid Partition Algorithm (and proof)

20. Last, we must make sure that the loop from lines 14 through 19 terminate correctly, as h is decremented down.
21. In lines 14 through 19, at the point when h is decremented down to $h = 1$, then at line 14, $I_1 + e \in \mathcal{I}_1$ (i.e., $I_1 + e$ is independent in M_1). If not, and $I_1 + e$ is dependent in M_1, then there is a unique circuit $C_1(I_1, e)$, and using the argument in lines 16, would imply we could find some $e' \in C_1(I_1, e) \setminus S_0 = C_1(I_1, e) \setminus E$ which obviously can't occur since E is the ground set.

Matroid Partition Algorithm (and proof)

20. Last, we must make sure that the loop from lines 14 through 19 terminate correctly, as h is decremented down.
21. In lines 14 through 19, at the point when h is decremented down to $h = 1$, then at line 14, $I_1 + e \in \mathcal{I}_1$ (i.e., $I_1 + e$ is independent in M_1). If not, and $I_1 + e$ is dependent in M_1, then there is a unique circuit $C_1(I_1, e)$, and using the argument in lines 16, would imply we could find some $e' \in C_1(I_1, e) \setminus S_0 = C_1(I_1, e) \setminus E$ which obviously can't occur since E is the ground set.
22. Therefore, decrement will terminate at some point at or before we hit $h = 1$.

if we reach some max indep set in some $h>1$, it will terminate before we reach $h=1$

Exercise:

Matroid Partition Algorithm (and proof)

20. Last, we must make sure that the loop from lines 14 through 19 terminate correctly, as h is decremented down.
21. In lines 14 through 19, at the point when h is decremented down to $h = 1$, then at line 14, $I_1 + e \in \mathcal{I}_1$ (i.e., $I_1 + e$ is independent in M_1). If not, and $I_1 + e$ is dependent in M_1, then there is a unique circuit $C_1(I_1, e)$, and using the argument in lines 16, would imply we could find some $e' \in C_1(I_1, e) \setminus S_0 = C_1(I_1, e) \setminus E$ which obviously can't occur since E is the ground set.
22. Therefore, decrement will terminate at some point at or before we hit $h = 1$.
23. This completes the algorithm, and the algorithmic proof.

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \dots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{5}$$

where r_i is the rank function of M_i.

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \dots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{5}$$

where r_i is the rank function of M_i.

- Now, if all matroids are the same $M_i = M$ for all i, we get condition

$$|A| \leq kr(A) \;\; \forall A \subseteq E \tag{6}$$

Matroid Partition Problem

Theorem 4.1

Let M_i be a collection of k matroids as described. Then, a set $I \subseteq E$ can be partitioned into k subsets $I_i, i = 1 \ldots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid i, if and only if, for all $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{5}$$

where r_i is the rank function of M_i.

- Now, if all matroids are the same $M_i = M$ for all i, we get condition
$$|A| \leq kr(A) \;\; \forall A \subseteq E \tag{6}$$
- But considering vector of all ones $\mathbf{1} \in \mathbb{R}_+^E$, this is the same as
$$\frac{1}{k}\mathbf{1}(A) \leq r(A) \;\; \forall A \subseteq E \tag{7}$$

Matroid Partition - Flow solution when $M = M_i, \forall i$

- It extends partition $(I_i : i \in J)$ of a proper subset of E into k independent sets, to such a partitioning of a larger subset.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- It extends partition $(I_i : i \in J)$ of a proper subset of E into k independent sets, to such a partitioning of a larger subset.
- At each step, we construct an auxiliary digraph graph G for this problem.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- It extends partition $(I_i : i \in J)$ of a proper subset of E into k independent sets, to such a partitioning of a larger subset.
- At each step, we construct an auxiliary digraph graph G for this problem.
- Vertex set is $E \cup \{s, t\}$ where s source and t sink are new nodes.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- It extends partition $(I_i : i \in J)$ of a proper subset of E into k independent sets, to such a partitioning of a larger subset.
- At each step, we construct an auxiliary digraph graph G for this problem.
- Vertex set is $E \cup \{s, t\}$ where s source and t sink are new nodes.
- Create directed edge (s, e) for all $e \in E$ such that $e \notin \cup_{i \in J} I_i$. That is, any element not yet in one of the independent sets.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- It extends partition $(I_i : i \in J)$ of a proper subset of E into k independent sets, to such a partitioning of a larger subset.
- At each step, we construct an auxiliary digraph graph G for this problem.
- Vertex set is $E \cup \{s, t\}$ where s source and t sink are new nodes.
- Create directed edge (s, e) for all $e \in E$ such that $e \notin \cup_{i \in J} I_i$. That is, any element not yet in one of the independent sets.
- Create directed edge (e, t) for all $e \in E$ such that $\exists i \in J$ with $e \notin I_i$ **and** $I_i + e \in \mathcal{I}$. I.e., we add this edge (e, t) if there is some independent set I_i that remains independent if e is added to it.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- It extends partition $(I_i : i \in J)$ of a proper subset of E into k independent sets, to such a partitioning of a larger subset.
- At each step, we construct an auxiliary digraph graph G for this problem.
- Vertex set is $E \cup \{s, t\}$ where s source and t sink are new nodes.
- Create directed edge (s, e) for all $e \in E$ such that $e \notin \cup_{i \in J} I_i$. That is, any element not yet in one of the independent sets.
- Create directed edge (e, t) for all $e \in E$ such that $\exists i \in J$ with $e \notin I_i$ **and** $I_i + e \in \mathcal{I}$. I.e., we add this edge (e, t) if there is some independent set I_i that remains independent if e is added to it.
- Add directed edge (e, f) for any distinct $e, f \in E$ such that $I_i + e \notin \mathcal{I}$ and $f \in C(I_i, e)$ for some i. That is, we add an edge (e, f) where e directs **to** the elements of a (nec. unique) circuit that is potentially created when e is added to I_i for some i.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- Therefore, incoming edges to e are either from source s, or from some other node that created a circuit in some I_i.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- Therefore, incoming edges to e are either from source s, or from some other node that created a circuit in some I_i.
- Outgoing edges from e are either to t, or are to nodes in the circuit created by e when it was added to some I_i.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- Therefore, incoming edges to e are either from source s, or from some other node that created a circuit in some I_i.
- Outgoing edges from e are either to t, or are to nodes in the circuit created by e when it was added to some I_i.
- So the outgoing edges from e either: 1) correspond to an independent set e may be added to, or 2) are to the circuit elements created when e is added to an independent set.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- Therefore, incoming edges to e are either from source s, or from some other node that created a circuit in some I_i.
- Outgoing edges from e are either to t, or are to nodes in the circuit created by e when it was added to some I_i.
- So the outgoing edges from e either: 1) correspond to an independent set e may be added to, or 2) are to the circuit elements created when e is added to an independent set.
- If the shortest path is $S = (s, e, t)$ then we can add e to some independent set and it is still independent.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- Therefore, incoming edges to e are either from source s, or from some other node that created a circuit in some I_i.
- Outgoing edges from e are either to t, or are to nodes in the circuit created by e when it was added to some I_i.
- So the outgoing edges from e either: 1) correspond to an independent set e may be added to, or 2) are to the circuit elements created when e is added to an independent set.
- If the shortest path is $S = (s, e, t)$ then we can add e to some independent set and it is still independent.
- If the shortest path is $S = (s, e, f, t)$ then we can add e to some I_1, create a circuit, but that gets broken when we remove f from that circuit rendering I_1 once again independent, but then there must be some other I_2 that f can be added to w/o making I_2 independent. Thus, the new independent sets are $I_1 + e - f$ and $I_2 + f$, thus we are making progress.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- Note that $I_1 \neq I_2$ since the edge (f, t) meant that we originally had $f \notin I_2$.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- Note that $I_1 \neq I_2$ since the edge (f, t) meant that we originally had $f \notin I_2$.
- If the shortest path is $S = (s, e, f_1, f_2, t)$ then we can:

Matroid Partition - Flow solution when $M = M_i, \forall i$

- Note that $I_1 \neq I_2$ since the edge (f, t) meant that we originally had $f \notin I_2$.
- If the shortest path is $S = (s, e, f_1, f_2, t)$ then we can:
  1. add e to some I_1, thus making a circuit C_1 due to edge (e, f_1).

Matroid Partition - Flow solution when $M = M_i, \forall i$

- Note that $I_1 \neq I_2$ since the edge (f, t) meant that we originally had $f \notin I_2$.
- If the shortest path is $S = (s, e, f_1, f_2, t)$ then we can:
  1. add e to some I_1, thus making a circuit C_1 due to edge (e, f_1).
  2. subtract f_1 from I_1, eliminating the circuit C_1.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- Note that $I_1 \neq I_2$ since the edge (f, t) meant that we originally had $f \notin I_2$.
- If the shortest path is $S = (s, e, f_1, f_2, t)$ then we can:
  1. add e to some I_1, thus making a circuit C_1 due to edge (e, f_1).
  2. subtract f_1 from I_1, eliminating the circuit C_1.
  3. add f_1 to some I_2, thus making a circuit C_2 due to edge (f_1, f_2).

Matroid Partition - Flow solution when $M = M_i, \forall i$

- Note that $I_1 \neq I_2$ since the edge (f, t) meant that we originally had $f \notin I_2$.
- If the shortest path is $S = (s, e, f_1, f_2, t)$ then we can:
  1. add e to some I_1, thus making a circuit C_1 due to edge (e, f_1).
  2. subtract f_1 from I_1, eliminating the circuit C_1.
  3. add f_1 to some I_2, thus making a circuit C_2 due to edge (f_1, f_2).
  4. subtract f_2 from I_2, eliminating the circuit C_2.

Matroid Partition - Flow solution when $M = M_i, \forall i$

- Note that $I_1 \neq I_2$ since the edge (f, t) meant that we originally had $f \notin I_2$.
- If the shortest path is $S = (s, e, f_1, f_2, t)$ then we can:
  1. add e to some I_1, thus making a circuit C_1 due to edge (e, f_1).
  2. subtract f_1 from I_1, eliminating the circuit C_1.
  3. add f_1 to some I_2, thus making a circuit C_2 due to edge (f_1, f_2).
  4. subtract f_2 from I_2, eliminating the circuit C_2.
  5. add f_2 to some I_3, not making a circuit due to edge (f_2, t).

  thus making progress.

Flow solution theorem

Thus, we have outlined the proof of one direction in the following theorem. When all matroids are the same $\forall i, M_i = M$ for some matroid, we have:

Theorem 4.2

There is an (s, t) path in the aforementioned graph iff the set of independent sets $(I_i : i \in J)$ can be grown by one element and still be a partition of some subset of E.

The other direction can be shown as a consequence of Theorem 4.1. Exercise

Problem To Solve

In particular, we will solve the following problem:

- Given a matroid $M = (E, \mathcal{I})$ along with an independence testing oracle (i.e., for any $A \subseteq E$, tells us if $A \in \mathcal{I}$ or not), and a vector $x \in \mathcal{R}_+^E$;
- find: a maximizing $y \in P_{\text{ind. set}}$ with $y \leq x$, and moreover (as a byproduct of the algorithm), express y as a convex combination of incidence vectors of independent sets in M, and also return a set $A \subseteq E$ that satisfies $y(E) = r_M(A) + x(E \setminus A)$. *Of course, for any such y we must have that $y(E) \leq r(A) + x(E \setminus A)$.*
- By the above theorem, the existence of such an A will certify that $x(E)$ is maximal in $P_{\text{ind. set}}$, minimal in terms of $f(A) = r_M(A) - x(A)$, and thus most violated in terms of A.
- This can also be used to test membership in $P_{\text{ind. set}}$ (i.e., if $y = x$) depending on the sign of f.
- This will also run in polynomial time.

Idea of the algorithm

- We build up y from the ground up.

Idea of the algorithm

- We build up y from the ground up.
- We keep a family of independent sets $(I_i : i \in J)$ and coefficients $(\lambda_i : i \in J)$ such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.

Idea of the algorithm

- We build up y from the ground up.
- We keep a family of independent sets $(I_i : i \in J)$ and coefficients $(\lambda_i : i \in J)$ such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- We gradually build up y by adding new independent sets (and augmenting J), adding to the existing independent sets, and adjusting coefficients.

Idea of the algorithm

- We build up y from the ground up.
- We keep a family of independent sets $(I_i : i \in J)$ and coefficients $(\lambda_i : i \in J)$ such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- We gradually build up y by adding new independent sets (and augmenting J), adding to the existing independent sets, and adjusting coefficients.
- and the way these additions are done is via solutions to a max-flow problem in an associated flow-graph (which we'll describe).

Idea of the algorithm

- We build up y from the ground up.
- We keep a family of independent sets $(I_i : i \in J)$ and coefficients $(\lambda_i : i \in J)$ such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- We gradually build up y by adding new independent sets (and augmenting J), adding to the existing independent sets, and adjusting coefficients.
- and the way these additions are done is via solutions to a max-flow problem in an associated flow-graph (which we'll describe).
- Each update will, of course, ensure that $y \in P_{\text{ind. set}}$, but also we'll keep $y \leq x$.

Associated digraph for polyhedra membership

- Define associated digraph G as follows.

Associated digraph for polyhedra membership

- Define associated digraph G as follows.
- Vertices of G, $V(G) = E \cup \{s, t\}$ where s, t are distinct elements not in E.

Associated digraph for polyhedra membership

- Define associated digraph G as follows.
- Vertices of G, $V(G) = E \cup \{s, t\}$ where s, t are distinct elements not in E.
- Create a directed edge (s, e) for all $e \in E$ such that $y(e) < x(e)$.

Associated digraph for polyhedra membership

- Define associated digraph G as follows.
- Vertices of G, $V(G) = E \cup \{s, t\}$ where s, t are distinct elements not in E.
- Create a directed edge (s, e) for all $e \in E$ such that $y(e) < x(e)$. Intuitively, y is our current measure of an "independence" of sorts, and any e s.t. $y(e) < x(e)$ is not yet "independent."

Associated digraph for polyhedra membership

- Define associated digraph G as follows.
- Vertices of G, $V(G) = E \cup \{s, t\}$ where s, t are distinct elements not in E.
- Create a directed edge (s, e) for all $e \in E$ such that $y(e) < x(e)$. Intuitively, y is our current measure of an "independence" of sorts, and any e s.t. $y(e) < x(e)$ is not yet "independent." (compare with: any $e \notin \cup_i I_i$ from before).

Associated digraph for polyhedra membership

- Define associated digraph G as follows.
- Vertices of G, $V(G) = E \cup \{s, t\}$ where s, t are distinct elements not in E.
- Create a directed edge (s, e) for all $e \in E$ such that $y(e) < x(e)$. Intuitively, y is our current measure of an "independence" of sorts, and any e s.t. $y(e) < x(e)$ is not yet "independent." (compare with: any $e \notin \cup_i I_i$ from before).
- Create directed edge (e, t) for all $e \in E$ such that $\exists i \in J$ with $e \notin I_i$ **and** $I_i + e \in \mathcal{I}$. I.e., we add this edge (e, t) if there is some independent set I_i that remains independent if e is added to it.

Associated digraph for polyhedra membership

- Define associated digraph G as follows.
- Vertices of G, $V(G) = E \cup \{s, t\}$ where s, t are distinct elements not in E.
- Create a directed edge (s, e) for all $e \in E$ such that $y(e) < x(e)$. Intuitively, y is our current measure of an "independence" of sorts, and any e s.t. $y(e) < x(e)$ is not yet "independent." (compare with: any $e \notin \cup_i I_i$ from before).
- Create directed edge (e, t) for all $e \in E$ such that $\exists i \in J$ with $e \notin I_i$ **and** $I_i + e \in \mathcal{I}$. I.e., we add this edge (e, t) if there is some independent set I_i that remains independent if e is added to it.
- Add directed edge (e, f) for any distinct $e, f \in E$ such that $I_i + e \notin \mathcal{I}$ and $f \in C(I_i, e)$ for some i. That is, we add an edge (e, f) where e directs **to** the elements of a (nec. unique) circuit that is potentially created when e is added to I_i for some i.

Associated digraph for polyhedra membership

- Define associated digraph G as follows.
- Vertices of G, $V(G) = E \cup \{s, t\}$ where s, t are distinct elements not in E.
- Create a directed edge (s, e) for all $e \in E$ such that $y(e) < x(e)$. Intuitively, y is our current measure of an "independence" of sorts, and any e s.t. $y(e) < x(e)$ is not yet "independent." (compare with: any $e \notin \cup_i I_i$ from before).
- Create directed edge (e, t) for all $e \in E$ such that $\exists i \in J$ with $e \notin I_i$ **and** $I_i + e \in \mathcal{I}$. I.e., we add this edge (e, t) if there is some independent set I_i that remains independent if e is added to it.
- Add directed edge (e, f) for any distinct $e, f \in E$ such that $I_i + e \notin \mathcal{I}$ and $f \in C(I_i, e)$ for some i. That is, we add an edge (e, f) where e directs **to** the elements of a (nec. unique) circuit that is potentially created when e is added to I_i for some i.
- The algorithm starts with $y = 0$, $J = \{0\}$, $I_0 = \emptyset$, and $\lambda_0 = 1$.

Augmenting path theorem

Theorem 5.1

If there is a directed path from s to t in G, then there exists $y' \in P$ with $y \leq y' \leq x$, with $y'(E) \geq y(E)$. If there is no such path, then there exists a set $A \subseteq E$ s.t. $y(A) = r(A)$ and $y(E \setminus A) = x(E \setminus A)$.

We will prove this next time.

Augmenting path theorem consequences

Corollary 5.2

For any $x \in \mathbb{R}_+^E$, we have

$$\max\left(y(E) : y \leq x, y \in P_f\right) = \min\left(x(A) + f(E \setminus A) : A \subseteq E\right) \qquad (16)$$

Note: this was not used in the theorem above, rather it is a consequence!

Proof.

- First, any $y \in P$ with $y \leq x$, and any $A \subset E$, we have

$$y(E) = y(A) + y(E \setminus A) \leq r(A) + x(E \setminus A) \qquad (17)$$

as we have seen.

□

Augmenting path theorem consequences

Corollary 5.2

For any $x \in \mathbb{R}_+^E$, *we have*

$$\max\left(y(E) : y \leq x, y \in P_f\right) = \min\left(x(A) + f(E \setminus A) : A \subseteq E\right) \quad (16)$$

Note: this was not used in the theorem above, rather it is a consequence!

Proof.

- First, any $y \in P$ with $y \leq x$, and any $A \subset E$, we have

$$y(E) = y(A) + y(E \setminus A) \leq r(A) + x(E \setminus A) \quad (17)$$

as we have seen.

- So we need only find a y giving equality.

Augmenting path theorem consequences

Corollary 5.2

For any $x \in \mathbb{R}_+^E$, *we have*

$$\max\left(y(E) : y \leq x, y \in P_f\right) = \min\left(x(A) + f(E \setminus A) : A \subseteq E\right) \tag{16}$$

Note: this was not used in the theorem above, rather it is a consequence!

Proof.

- First, any $y \in P$ with $y \leq x$, and any $A \subset E$, we have

$$y(E) = y(A) + y(E \setminus A) \leq r(A) + x(E \setminus A) \tag{17}$$

  as we have seen.
- So we need only find a y giving equality.
- Choose any $y \in P$ such that $y \leq x$ and with $y(E)$ maximum.

Augmenting path theorem consequences

Corollary 5.2

For any $x \in \mathbb{R}_+^E$, *we have*

$$\max\left(y(E) : y \leq x, y \in P_f\right) = \min\left(x(A) + f(E \setminus A) : A \subseteq E\right) \tag{16}$$

Note: this was not used in the theorem above, rather it is a consequence!

Proof.

- First, any $y \in P$ with $y \leq x$, and any $A \subset E$, we have

$$y(E) = y(A) + y(E \setminus A) \leq r(A) + x(E \setminus A) \tag{17}$$

  as we have seen.
- So we need only find a y giving equality.
- Choose any $y \in P$ such that $y \leq x$ and with $y(E)$ maximum.
- Then there exists no such $y' \in P$ s.t. $y'(E) > y(E)$, and the digraph won't have a directed path from s to t (by the theorem).

□

Augmenting path theorem consequences

Corollary 5.2

For any $x \in \mathbb{R}_+^E$, we have

$$\max\left(y(E) : y \leq x, y \in P_f\right) = \min\left(x(A) + f(E \setminus A) : A \subseteq E\right) \tag{16}$$

Note: this was not used in the theorem above, rather it is a consequence!

Proof.

- First, any $y \in P$ with $y \leq x$, and any $A \subset E$, we have
$$y(E) = y(A) + y(E \setminus A) \leq r(A) + x(E \setminus A) \tag{17}$$
as we have seen.
- So we need only find a y giving equality.
- Choose any $y \in P$ such that $y \leq x$ and with $y(E)$ maximum.
- Then there exists no such $y' \in P$ s.t. $y'(E) > y(E)$, and the digraph won't have a directed path from s to t (by the theorem).
- Then, there is a set A such that $y(A) = r(A)$ and $y(E \setminus A) = x(E \setminus A)$, or that $y(E) = r(A) + x(E \setminus A)$, thus demonstrating equality.

□

Scratch Paper

Scratch Paper

Scratch Paper

Sources for Today's Lecture

- Jack Edmonds, "Matroid Partition", 1968.
- W. Cunningham, "Testing Membership in Matroid Polyhedra", 1984
- E. Lawler, "Matroid Intersection Algorithms", 1975.
- L. Schrijver, "Combinatorial Optimization", 2003.
- Krogdahl, "A Combinatorial Base for some Optimal Matroid Intersection Algorithms", 1974.