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

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

(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

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

(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:*

\[
\max\ \{y(E) : y \in \text{P}_{\text{ind. set}}(M), \ y \leq x\} = \min\ \{r_M(A) + x(E \setminus A) : A \subseteq E\}
\]

(2)

where \( r_M \) is the matroid rank function of some matroid.
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

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

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

Proof.

Exercise

And its rank function is Exercise.
Most violated inequality problem

Consider

\[ Pr = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \]  \hspace{1cm} (4)
Consider

\[ P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \]  \hspace{1cm} (4)

We saw before that \( P_r = P_{\text{ind. set}} \).
Consider

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

We saw before that \( P_r = P_{\text{ind. set}} \).

Suppose we have any \( x \in \mathbb{R}_+^E \) such that \( x \not\in P_r \).
Most violated inequality problem

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

- We saw before that \( P_r = P_{\text{ind. set.}} \).

- Suppose we have any \( x \in \mathbb{R}^E_+ \) such that \( x \not\in P_r \).

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

- We saw before that \( P_r = P_{\text{ind. set}} \).

- Suppose we have any \( x \in \mathbb{R}^E_+ \) such that \( x \not\in P_r \).

- 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.,
  \[ \max \left\{ x(A) - r_M(A) : A \subseteq E \right\} \]

- This corresponds to
  \[ \min \left\{ r_M(A) + x(E \setminus A) : A \subseteq E \right\} \]  
  since \( x \) is modular and \( x(E \setminus A) = x(E) - x(A) \).

Prof. Jeff Bilmes
EE595A/Spr 2011/Submodular Functions – Lecture 9 - April 29th, 2011
Most violated inequality problem

- Consider
  \[ P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \quad (4) \]

- 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.,
  \[ \max \left\{ x(A) - r_M(A) : A \subseteq E \right\} \]

- This corresponds to \( \min \left\{ r_M(A) + x(E \setminus A) : A \subseteq E \right\} \) since \( x \) is modular and \( x(E \setminus A) = x(E) - x(A) \).

- More importantly, \( \min \left\{ r_M(A) + x(E \setminus A) : A \subseteq E \right\} \) a form of submodular function minimization, namely
  \[ \min \left\{ r_M(A) - x(A) : A \subseteq E \right\} \] 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 $x \in \mathbb{R}^E_+$;
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 $x \in \mathbb{R}^E_+$;

- 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 $x \in \mathbb{R}^E_+$;

- 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 $x \in \mathbb{R}_+^E$;
- 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 $x \in \mathbb{R}^E_+$;

- 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 $y = \sum_{i \in J} \lambda_i 1_{I_i}$.
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 $y = \sum_{i \in J} \lambda_i 1_{I_i}$.
- 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 $(l_i : i \in J)$ and coefficients $(\lambda_i : i \in J)$ such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i 1_{l_i}$.
- 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 $(l_i : i \in J)$ and coefficients $(\lambda_i : i \in J)$ such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i 1_{l_i}$.
- 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 $(l_i : i \in J)$ and coefficients $(\lambda_i : i \in J)$ such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i 1_{l_i}$.
- 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 $M_1 = (V, I_1)$ and $M_2 = (V, I_2)$ and start with any $I \in I_1 \cap I_2$.
- Consider some $v_1 \notin \text{span}_1(I)$, so that $I + v_1 \in I_1$.
- If $I + v_1 \in I_2$, 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 $M_1$. 

Prof. Jeff Bilmes
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 $\nu_1 \notin \text{span}_1(I)$, so that $I + \nu_1 \in \mathcal{I}_1$.

If $I + \nu_1 \in \mathcal{I}_2$, then $\nu_1$ is “augmenting”, and we can augment $I$ to $I + \nu_1$ and still be independent in both $M_1$ and $M_2$.

If $I + \nu_1 \notin \mathcal{I}_2$, $\exists C_2(I, \nu_1)$ a circuit in $M_2$, and choosing $\nu_2 \in C_2(I, \nu_1)$ s.t. $\nu_2 \neq \nu_1$ leads to $I + \nu_1 - \nu_2$ which (because $\text{span}_2(I) = \text{span}(I + \nu_1 - \nu_2)$) is again independent in $M_2$. It is also independent in $M_1$.

Next choose a $\nu_3 \in \text{span}_1(I) - \text{span}_1(I - \nu_2)$ to recover what was lost in $I \cup \{\nu_1\}$ when we removed $\nu_2$ from it.
Matroid Intersection Algorithm Idea

- Consider two matroids $M_1 = (V, I_1)$ and $M_2 = (V, I_2)$ and start with any $I \in I_1 \cap I_2$.
- Consider some $v_1 \not\in \text{span}_1(I)$, so that $I + v_1 \in I_1$.
- If $I + v_1 \in I_2$, 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 $I + v_1 \not\in I_2$, then there exists a circuit $C_2(I, v_1)$ 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 $M_1 = (V, I_1)$ and $M_2 = (V, I_2)$ and start with any $I \in I_1 \cap I_2$.
- Consider some $v_1 \notin \text{span}_1(I)$, so that $I + v_1 \in I_1$.
- If $I + v_1 \in I_2$, 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 $I + v_1 \notin I_2$, $\exists C_2(I, v_1)$ 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 $I + v_1 \in I_1$, $v_1 \notin \text{span}_1(I)$, 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 $M_1$ and $M_2$ 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 $M_1$ and $M_2$ 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.
Let $I$ be an intersection of two matroids $M_1 = (E, I_1)$ and $M_2 = (E, I_2)$ (i.e., $I \in I_1 \cap I_2$).
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 $S = (e_1, e_2, \ldots, e_s)$ 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.
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 $S = (e_1, e_2, \ldots, e_s)$ 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. $I + e_1 \in \mathcal{I}_1$
2. For all even $i$, span$_2(I \oplus S_i) = \text{span}_2(I)$ which implies that $I \oplus S_i \in \mathcal{I}_2$.
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 $S = (e_1, e_2, \ldots, e_s)$ 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. $I + e_1 \in \mathcal{I}_1$
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 $I \ominus S_i \in \mathcal{I}_1$.
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. $I + e_1 \in \mathcal{I}_1$
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. $I$. 
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 $M_1 = (V, I_1)$ and $M_2 = (V, I_2)$
Matroid Intersection Algorithm

- Given two matroids $M_1 = (V, I_1)$ and $M_2 = (V, I_2)$
- Our goal is to find the maximum cardinality set $I$ such that $I \in I_1$ and $I \in I_2$. 

Algorithm 9.1: Alternating Path Matroid Intersection

1. Let $I$ be an arbitrary (including empty) independent set in two matroids $M_1$ and $M_2$;
2. while There exists an augmenting sequence $S$ do
3. $I \leftarrow I \setminus S$;

This can be made to run in $O(m_2R + mR^2c(m))$ 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).
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

1. Let $I$ be an arbitrary (including empty) independent set in two matroids $M_1$ and $M_2$;
2. while There exists an augmenting sequence $S$ do
3. $I \leftarrow I \ominus S$;

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

- Given two matroids $M_1 = (V, I_1)$ and $M_2 = (V, I_2)$
- Our goal is to find the maximum cardinality set $I$ such that $I \in I_1$ and $I \in I_2$.
- The algorithm described becomes:

**Algorithm 9.1: Alternating Path Matroid Intersection**

1. Let $I$ be an arbitrary (including empty) independent set in two matroids $M_1$ and $M_2$;
2. while There exists an augmenting sequence $S$ do
   3. $I \leftarrow I \oplus S$;

- 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).
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 $e_j \in C_1(I, e_i) \setminus \{e_i\}$.
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.
\{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? \[\_\_\_\_\_\_\_\_.\]
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 $S \subseteq I \ominus J$. 
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 $\text{span}_2(I) \subseteq \text{span}_2(I^*)$. 

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 $\text{span}_2(I) \subseteq \text{span}_2(I^*)$. 
Suppose $M_i = (E, I_i)$ 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, $I_i, i \in \{1, 2, \ldots, k\}$ where $I_i \in \mathcal{I}_i$. 
Suppose $M_i = (E, I_i)$ 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 $I_i \in I_i$.

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 $I_i, i = 1 \ldots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid $i$, if and only if, for all $A \subseteq I$

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

where $r_i$ is the rank function of $M_i$. 

Prof. Jeff Bilmes
EE595A/Spr 2011/Submodular Functions – Lecture 9 - April 29th, 2011
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 $A \subseteq I$

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

(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

$$|A| \leq kr(A) \ \forall A \subseteq E$$

(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 $A \subseteq I$

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

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

$$|A| \leq kr(A) \quad \forall A \subseteq E \quad (6)$$

- But considering vector of all ones $\mathbf{1} \in \mathbb{R}^E_+$, this is the same as

$$\frac{1}{k} \mathbf{1}(A) \leq r(A) \quad \forall A \subseteq E \quad (7)$$
Matroid Partition Problem and Submodular Function Minimization

- Recall definition of matroid polytope

\[ Pr = \left\{ y \in \mathbb{R}_+^E : y(A) \leq r(A) \text{ for all } A \subseteq E \right\} \quad (11) \]
Recall definition of matroid polytope

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

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

- 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).
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 $A \subseteq I$

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

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 $I_i$, $i = 1 \ldots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid $i$, if and only if, for all $A \subseteq I$

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

(5)

where $r_i$ is the rank function of $M_i$.

Proof.

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

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

(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 $I_i, i = 1 \ldots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid $i$, if and only if, for all $A \subseteq I$

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

(5)

where $r_i$ is the rank function of $M_i$.

Proof.

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

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

(8)

- and moreover for each independent set $I_i$, we have

$$|I_i \cap A| \leq r_i(A)$$

(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 $I_i, i = 1 \ldots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid $i$, if and only if, for all $A \subseteq I$

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

where $r_i$ is the rank function of $M_i$.

Proof.

which immediately gives:

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

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 $I_i$, $i = 1 \ldots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid $i$, if and only if, for all $A \subseteq I$

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

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 $I_i$, $i = 1 \ldots k$ where $l_i \in I_i$ is independent in matroid $i$, if and only if, for all $A \subseteq I$

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

(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 $I_i$, $i = 1 \ldots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid $i$, if and only if, for all $A \subseteq I$

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

(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 $I_i, i = 1 \ldots k$ where $I_i \in \mathcal{I}_i$ is independent in matroid $i$, if and only if, for all $A \subseteq I$

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

(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 $S_j$, there will be an associated $I_j(i)$. For simplicity, we will be calling them $I_i$.
   - $I_j(i) = I_i$ w.l.o.g., there might be duplicates, i.e. $\sqrt{j(i)} = j(i')$ for $i \neq i'$.
The algorithm starts with a set of $k$ empty sets, $\mathcal{I}_k = \emptyset$, $i = 1, 2, \ldots, k$. $J$ is our index set, so $J = \{1, 2, \ldots, k\}$.

We are going to create a sequence of subsets $(\mathcal{S}_0, \mathcal{S}_1, \ldots)$, starting with $\mathcal{S}_0 = E$, and the others are defined by the algorithm below.

Assume there is an $e \in E$ s.t. $e \notin \bigcup_i \mathcal{I}_i$ (otherwise we’re done).
Matroid Partition Algorithm (and proof)

1. The algorithm starts with a set of \( k \) empty sets, \( l_k = \emptyset, i = 1, 2, \ldots, k \). \( J \) 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. \( e \notin \bigcup_i l_i \) (otherwise we’re done).

4. If \( \forall i, |l_i| \geq r_i(E) \), then \( |E| \geq |\{e\} + \bigcup_i l_i| > \sum_i r_i(E) \) 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. $e \not\in \bigcup_i I_i$ (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 $S_1 \overset{\text{def}}{=} \text{span}_1(I_1) = S_0 \cap \text{span}_1(I_1 \cap S_0) = E \cap \text{span}_1(I_1 \cap E)$. 

$$\checkmark$$
Matroid Partition Algorithm (and proof)

1. The algorithm starts with a set of $k$ empty sets, $l_k = \emptyset, i = 1, 2, \ldots, k$. $J$ 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 $S_1 \overset{\text{def}}{=} \text{span}_1(I_1) = S_0 \cap \text{span}_1(I_1 \cap S_0) = E \cap \text{span}_1(I_1 \cap E)$.

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 $\forall i, e \notin I_i$).
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 $S_1 \overset{\text{def}}{=} \text{span}_1(I_1) = S_0 \cap \text{span}_1(I_1 \cap S_0) = E \cap \text{span}_1(I_1 \cap E)$.
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 $S_2 \overset{\text{def}}{=} S_1 \cap \text{span}_2(I_2 \cap S_1)$. $S_2 \subseteq S_1$.
Note $S_2 \subseteq S_1$. Moreover, we have $S_2 \subset S_1$ (proper) since
\begin{align*}
r_2(S_2) &= r_2(S_1 \cap \text{span}_2(I_2 \cap S_1)) \quad \text{by def.} \\
&\leq r_2(\text{span}_2(I_2 \cap S_1)) \quad \text{max. of rank} \\
&= r_2(I_2 \cap S_1) \\
&\leq |I_2 \cap S_1| < r_2(S_1) \quad \text{by assumption}
\end{align*}
and if the rank decreases, the sets can’t be equal.
\[ S_2 \subset S_1 \]
Note $S_2 \subseteq S_1$. Moreover, we have $S_2 \subset S_1$ (proper) since
\[ r_2(S_2) = r_2(S_1 \cap \text{span}_2(l_2 \cap S_1)) \]
\[ \leq r_2(\text{span}_2(l_2 \cap S_1)) \]
\[ = r_2(l_2 \cap S_1) \]
\[ \leq |l_2 \cap S_1| < r_2(S_1) \]
and if the rank decreases, the sets can’t be equal.

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

\[ \Rightarrow |S_2| > \bigcap_{i \neq j} r_i(S_i) \]
Matroid Partition Algorithm (and proof)

8 Note $S_2 \subseteq S_1$. Moreover, we have $S_2 \subset S_1$ (proper) since
\[
    r_2(S_2) = r_2(S_1 \cap \text{span}_2(l_2 \cap S_1)) \quad (12)
\]
\[
    \leq r_2(\text{span}_2(l_2 \cap S_1)) \quad (13)
\]
\[
    = r_2(l_2 \cap S_1) \quad (14)
\]
\[
    \leq |l_2 \cap S_1| < r_2(S_1) \quad (15)
\]
and if the rank decreases, the sets can’t be equal.

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

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

8. Note $S_2 \subseteq S_1$. Moreover, we have $S_2 \subset S_1$ (proper) since
\[
  r_2(S_2) = r_2(S_1 \cap \text{span}_2(I_2 \cap S_1)) \quad (12)
\]
\[
  \leq r_2(\text{span}_2(I_2 \cap S_1)) \quad (13)
\]
\[
  = r_2(I_2 \cap S_1) \quad (14)
\]
\[
  \leq |I_2 \cap S_1| < r_2(S_1) \quad (15)
\]

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 $|S_j| \geq |\{e\} \cup \bigcup_i (I_i \cap S_j)| > \sum_i r_i(S_j)$ 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 $S_{j+1} \overset{\text{def}}{=} S_j \cap \text{span}_{j+1}(I_{j+1} \cap S_j)$.

11. And we have $S_{j+1} \subset S_j$. By same argument as eqns 12-15.
Note $S_2 \subseteq S_1$. Moreover, we have $S_2 \subset S_1$ (proper) since
\[ r_2(S_2) = r_2(S_1 \cap \text{span}_2(l_2 \cap S_1)) \]
\[ \leq r_2(\text{span}_2(l_2 \cap S_1)) \]
\[ = r_2(l_2 \cap S_1) \]
\[ \leq |l_2 \cap S_1| < r_2(S_1) \]
and if the rank decreases, the sets can’t be equal.

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 $\forall i$, $e \not\in I_i$).

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 $S_{j+1} \overset{\text{def}}{=} S_j \cap \text{span}_{j+1}(I_{j+1} \cap S_j)$.

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

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.
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:

13. **Case 1:** 
   - If $I_h + e \in I_h$, 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.

14. **Case 2:**
   - If, on the other hand, $I_h + e \notin M_h$ then there is a (necessarily unique) circuit $C_h(I_h, e)$ created in $M_h$ when adding $e$ to $I_h$.

15. 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 $S_h = S_h - 1 \cap \text{span}_{h}(I_h \cap S_h - 1)$, this also implies $C_h(I_h, e) \subseteq S_h$, which is impossible since $e \notin S_h$ (from line 13 or 17).

16. Thus, we choose 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 $j < h$.
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:

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 $I_h$. 

Next, decrement $h \leftarrow h - 1$, update $e \leftarrow e'$, and GOTO line 14.
Due to the strict monotone decreasing, at some point we must get to an $h$ such that $e \not\in S_h$. Two things can then happen:

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.

If, on the other hand, $I_h + e \not\in M_h$ then there is a (nec. unique) circuit $C_h(I_h, e)$ created in $M_h$ when adding $e$ to $I_h$.

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 $S_h \overset{\text{def}}{=} S_{h-1} \cap \text{span}_h(I_h \cap S_{h-1})$, this also implies $C_h(I_h, e) \subseteq S_h$, which is impossible since $e \not\in S_h$ (from line 13 or 17).
Due to the strict monotone decreasing, at some point we must get to an $h$ such that $e \not\in S_h$. Two things can then happen:

If $I_h + e \in I_h$ 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.

If, on the other hand, $I_h + e \not\in M_h$ then there is a (nec. unique) circuit $C_h(I_h, e)$ created in $M_h$ when adding $e$ to $I_h$.

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 $S_h \overset{\text{def}}{=} S_{h-1} \cap \text{span}_h(I_h \cap S_{h-1})$, this also implies $C_h(I_h, e) \subseteq S_h$, which is impossible since $e \not\in S_h$ (from line 13 or 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 $j < h$. 

---

13  14  15  16  17
Due to the strict monotone decreasing, at some point we must get to an $h$ such that $e \not\in S_h$. Two things can then happen:

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.

If, on the other hand, $I_h + e \not\in M_h$ then there is a (nec. unique) circuit $C_h(I_h, e)$ created in $M_h$ when adding $e$ to $I_h$.

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 $S_h \overset{\text{def}}{=} S_{h-1} \cap \text{span}_h(I_h \cap S_{h-1})$, this also implies $C_h(I_h, e) \subseteq S_h$, which is impossible since $e \not\in S_h$ (from line 13 or 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 $j < h$.

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 $S_h \overset{\text{def}}{=} S_{h-1} \cap \text{span}_h(I_h \cap S_{h-1})$, 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.
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.
Last, we must make sure that the loop from lines 14 through 19 terminate correctly, as $h$ is decremented down.

In lines 14 through 19, at the point when $h$ is decremented down to $h = 1$, then at line 14, $I_1 + e \in I_1$ (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.

Therefore, decrement will terminate at some point at or before we hit $h = 1$. 

\[ \text{Exercises} \]
Last, we must make sure that the loop from lines 14 through 19 terminate correctly, as $h$ is decremented down.

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.

Therefore, decrement will terminate at some point at or before we hit $h = 1$.

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 $I_i, i = 1 \ldots k$ where $l_i \in I_i$ 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 $M_i$. 
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 $A \subseteq I$

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

$$|A| \leq kr(A) \ \forall A \subseteq E$$  \hspace{1cm} (6)
Matroid Partition Problem

Theorem 4.1

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

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

(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

$$|A| \leq kr(A) \ \forall A \subseteq E$$

(6)

- But considering vector of all ones $1 \in \mathbb{R}^E_+$, this is the same as

$$\frac{1}{k}1(A) \leq r(A) \ \forall A \subseteq E$$

(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 $e \notin \bigcup_{i \in J} I_i$. 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.
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 \(e \notin \bigcup_{i \in J} I_i\). 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 $I_i$. 

---

Prof. Jeff Bilmes
EE595A/Spr 2011/Submodular Functions – Lecture 9 - April 29th, 2011
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 $l_1 \neq l_2$ since the edge $(f, t)$ meant that we originally had $f \not\in l_2$. 
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 $f \not\in I_2$.
- 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.
Matroid Partition - Flow solution when $M = M_i, \forall i$

- Note that $l_1 \neq l_2$ since the edge $(f, t)$ meant that we originally had $f \notin l_2$.

- If the shortest path is $S = (s, e, f_1, f_2, t)$ then we can:
  1. add $e$ to some $l_1$, thus making a circuit $C_1$ due to edge $(e, f_1)$.  
  2. subtract $f_1$ from $l_1$, eliminating the circuit $C_1$.
  3. add $f_1$ to some $l_2$, thus making a circuit $C_2$ due to edge $(f_1, f_2)$.
  4. subtract $f_2$ from $l_2$, eliminating the circuit $C_2$.
  5. add $f_2$ to some $l_3$, not making a circuit due to edge $(f_2, t)$.
Note that $l_1 \neq l_2$ since the edge $(f, t)$ meant that we originally had $f \notin l_2$.

If the shortest path is $S = (s, e, f_1, f_2, t)$ then we can:

1. add $e$ to some $l_1$, thus making a circuit $C_1$ due to edge $(e, f_1)$.
2. subtract $f_1$ from $l_1$, 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 $l_1 \neq l_2$ since the edge $(f, t)$ meant that we originally had $f \not\in I_2$.

- If the shortest path is $S = (s, e, f_1, f_2, t)$ then we can:
  1. add $e$ to some $l_1$, thus making a circuit $C_1$ due to edge $(e, f_1)$.
  2. subtract $f_1$ from $l_1$, eliminating the circuit $C_1$.
  3. add $f_1$ to some $l_2$, thus making a circuit $C_2$ due to edge $(f_1, f_2)$.
  4. subtract $f_2$ from $l_2$, 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.
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 \( x \in \mathbb{R}_+^E \);
- find: a maximizing \( y \in \text{P}_{\text{ind. set}} \) 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 \( \text{P}_{\text{ind. set}} \), 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 \( \text{P}_{\text{ind. set}} \) (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 ($l_i : i \in J$) and coefficients ($\lambda_i : i \in J$) such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i 1_{l_i}$. 
Idea of the algorithm

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

We keep a family of independent sets $(l_i : i \in J)$ and coefficients $(\lambda_i : i \in J)$ such that $\sum_{i \in J} \lambda_i = 1$ and $y = \sum_{i \in J} \lambda_i 1_{l_i}$.

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 $y = \sum_{i \in J} \lambda_i 1_{l_i}$. 
- We gradually build up $y$ by adding new independent sets (and augmenting $J$), adding to the existing independent sets, and adjusting coefficients.
- 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 $y \in \mathcal{P}_{\text{ind. set}}$, but also we’ll keep $y \leq x$. 

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 $e/\in \bigcup_i I_i$ from before).
- Create directed edge $(e, t)$ for all $e \in E$ such that $\exists i \in J$ with $e/\in I_i$ and $I_i + e \in I$.
- 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 $\lambda_0 = 1$. 

Associated digraph for polyhedra membership
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 $E$. 

Define associated digraph $G$ as follows.

Vertices of $G$, $V(G) = E \cup \{s, t\}$ where $s, t$ are distinct elements not in $E$. 

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 $e/\in \bigcup_i I_i$ from before).

Create directed edge $(s, e)$ for all $e \in E$ such that $y(e) < x(e)$.

Create directed edge $(e, t)$ for all $e \in E$ such that $\exists i \in J$ with $e/\in I_i$ and $I_i + e \in I$. 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 $I_i + e /\in I_i$ 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 $\lambda_0 = 1$. 

Prof. Jeff Bilmes
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 $e \not\in \bigcup_i I_i$ from before).

Create directed edge $(e, t)$ for all $e \in E$ such that $\exists i \in J$ with $e \not\in I_i$ and $I_i + e \in I$.

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 $\lambda_0 = 1$. 
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 $e \notin \bigcup_i I_i$ from before).
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 $e \notin \bigcup_i I_i$ 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.
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 $e \notin \bigcup_i I_i$ 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 $\lambda_0 = 1$. 

---

**Prof. Jeff Bilmes**

EE595A/Spr 2011/Submodular Functions – Lecture 9 - April 29th, 2011
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 $e \notin \bigcup_i I_i$ 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.
Corollary 5.2

For any \( x \in \mathbb{R}^E_+ \), we have

\[
\max \left( y(E) : y \leq x, y \in P_f \right) = \min \left( x(A) + f(E \setminus A) : A \subseteq E \right)
\] (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 \( A \subseteq E \), we have

\[
y(E) = y(A) + y(E \setminus A) \leq r(A) + x(E \setminus A)
\] (17)

as we have seen.
Augmenting path theorem consequences

Corollary 5.2

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

$$\max (y(E) : y \leq x, y \in P_f) = \min (x(A) + f(E \setminus A) : A \subseteq E)$$  \hfill (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 $A \subseteq E$, we have

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

\[
\max (y(E) : y \leq x, y \in P_f) = \min (x(A) + f(E \setminus A) : A \subseteq E)
\] (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

\[
y(E) = y(A) + y(E \setminus A) \leq r(A) + x(E \setminus A)
\] (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

\[
\max (y(E) : y \leq x, y \in P_f) = \min (x(A) + f(E \setminus A) : A \subseteq E)
\] (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 \( A \subseteq E \), we have

\[
y(E) = y(A) + y(E \setminus A) \leq r(A) + x(E \setminus A)
\] (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

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

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.

- 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.
Sources for Today’s Lecture

- W. Cunningham, “Testing Membership in Matroid Polyhedra”, 1984