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 10 - May 4th, 2011

Announcements

- On Final projects. **One** single page final project proposals (revision one) are due this 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, say, lead to a NIPS paper in June and be related to submodularity.

Most violated inequality problem

- Consider

$$P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \tag{1}$$

Most violated inequality problem

- Consider
$$P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \quad (1)$$
- We saw before that $P_r = P_{\text{ind. set}}$.

Most violated inequality problem

- Consider

$$P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \tag{1}$$

- 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
$$P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \tag{1}$$
- 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
$$P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \tag{1}$$
- 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
$$P_r = \left\{ x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E \right\} \quad (1)$$
- 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 $y(E)$ is maximal in $P_{\text{ind. set}}$, A is minimal in terms of $f(A) \stackrel{\text{def}}{=} r_M(A) - x(A)$ (thus most violated).

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 $y(E)$ is maximal in $P_{\text{ind. set}}$, A is minimal in terms of $f(A) \stackrel{\text{def}}{=} r_M(A) - x(A)$ (thus most violated).
- This can also be used to test membership in $P_{\text{ind. set}}$ (i.e., if $y = x$) depending on the sign of f at 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 $y(E)$ is maximal in $P_{\text{ind. set}}$, A is minimal in terms of $f(A) \stackrel{\text{def}}{=} r_M(A) - x(A)$ (thus most violated).
- This can also be used to test membership in $P_{\text{ind. set}}$ (i.e., if $y = x$) depending on the sign of f at A.
- This will also run in polynomial time.

Idea of the algorithm

- Given $x \in \mathbb{R}_+$, we build up y from the ground up, ensuring that $y \leq x$, and starting with $y = 0$..

Idea of the algorithm

- Given $x \in \mathbb{R}_+$, we build up y from the ground up, ensuring that $y \leq x$, and starting with $y = 0$..
- 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

- Given $x \in \mathbb{R}_+$, we build up y from the ground up, ensuring that $y \leq x$, and starting with $y = 0$..
- 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 $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- Therefore, $y \in P_{\text{ind. set}}$.
- 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

- Given $x \in \mathbb{R}_+$, we build up y from the ground up, ensuring that $y \leq x$, and starting with $y = 0$..
- 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 $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- Therefore, $y \in P_{\text{ind. set}}$.
- 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

- Given $x \in \mathbb{R}_+$, we build up y from the ground up, ensuring that $y \leq x$, and starting with $y = 0$..
- 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 $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- Therefore, $y \in P_{\text{ind. set}}$.
- 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

- Given $x \in \mathbb{R}_+$, we build up y from the ground up, ensuring that $y \leq x$, and starting with $y = 0$..
- 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 $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- Therefore, $y \in P_{\text{ind. set}}$.
- 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 has taken us a few lectures to fully develop this algorithm, today we will probably finish it.

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, $I_i, i \in \{1, 2, \ldots, k\}$ 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, $I_i, i \in \{1, 2, \dots, k\}$ 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 2.1

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 $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{2}$$

where r_i is the rank function of M_i.

Matroid Partition Problem

Theorem 2.1

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 $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{2}$$

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

$$|A| \leq kr(A) \;\; \forall A \subseteq E \tag{3}$$

Matroid Partition Problem

Theorem 2.1

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 $A \subseteq I$

$$|A| \leq \sum_{i=1}^{k} r_i(A) \tag{2}$$

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
$$|A| \leq kr(A) \;\; \forall A \subseteq E \tag{3}$$
- But considering vector of all ones $\mathbf{1} \in \mathbb{R}_+^E$, this is the same as
$$\frac{1}{k}\mathbf{1}(A) \leq r(A) \;\; \forall A \subseteq E \tag{4}$$

Matroid Partition Problem and Submodular Function Minimization

- Recall definition of matroid polytope

$$P_r = \left\{ y \in \mathbb{R}_+^E : y(A) \leq r(A) \text{ for all } A \subseteq E \right\} \quad (5)$$

Matroid Partition Problem and Submodular Function Minimization

- Recall definition of matroid polytope
$$P_r = \left\{ y \in \mathbb{R}_+^E : y(A) \leq r(A) \text{ for all } A \subseteq E \right\} \quad (5)$$
- 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
$$P_r = \left\{ y \in \mathbb{R}_+^E : y(A) \leq r(A) \text{ for all } A \subseteq E \right\} \quad (5)$$
- 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
$$P_r = \left\{ y \in \mathbb{R}_+^E : y(A) \leq r(A) \text{ for all } A \subseteq E \right\} \tag{5}$$
- 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 $\sum_i r_i(A) - \mathbf{1}(A)$, 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 - 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 since overall, e is added.

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.

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.
- Here, $I_1 \neq I_2$, and $I_2 \neq I_3$, but could have $I_1 = I_3$ Exercise:

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 3.1

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 2.1. 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 $y(E)$ is maximal in $P_{\text{ind. set}}$, A is minimal in terms of $f(A) \stackrel{\text{def}}{=} r_M(A) - x(A)$ (thus most violated).
- This can also be used to test membership in $P_{\text{ind. set}}$ (i.e., if $y = x$) depending on the sign of f at A.
- This will also run in polynomial time.

Idea of the algorithm

- Given $x \in \mathbb{R}_+$, we build up y from the ground up, ensuring that $y \leq x$, and starting with $y = 0$..

Idea of the algorithm

- Given $x \in \mathbb{R}_+$, we build up y from the ground up, ensuring that $y \leq x$, and starting with $y = 0$..
- 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

- Given $x \in \mathbb{R}_+$, we build up y from the ground up, ensuring that $y \leq x$, and starting with $y = 0$..
- 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 $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- Therefore, $y \in P_{\text{ind. set}}$.
- 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

- Given $x \in \mathbb{R}_+$, we build up y from the ground up, ensuring that $y \leq x$, and starting with $y = 0$..
- 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 $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- Therefore, $y \in P_{\text{ind. set}}$.
- 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

- Given $x \in \mathbb{R}_+$, we build up y from the ground up, ensuring that $y \leq x$, and starting with $y = 0$..
- 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 $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$.
- Therefore, $y \in P_{\text{ind. set}}$.
- 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 "saturated"

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 "saturated" (compare with: any $e \notin \cup_i I_i$ from matroid partition case).

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 "saturated" (compare with: any $e \notin \cup_i I_i$ from matroid partition case).
- 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 "saturated" (compare with: any $e \notin \cup_i I_i$ from matroid partition case).
- 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 "saturated" (compare with: any $e \notin \cup_i I_i$ from matroid partition case).
- 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

- Thus, we consider $x \in \mathbb{R}_+$.

Augmenting path theorem

- Thus, we consider $x \in \mathbb{R}_+$.
- We've constructed the aforementioned s, t graph G as previously mentioned, where for each $e \in E$, we've got a node in G, along with additional nodes (and edges) s, t.

Augmenting path theorem

- Thus, we consider $x \in \mathbb{R}_+$.
- We've constructed the aforementioned s, t graph G as previously mentioned, where for each $e \in E$, we've got a node in G, along with additional nodes (and edges) s, t.
- We maintaine $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i} \leq x$ and thus $y \in P_{\text{ind. set}}$.

Augmenting path theorem

- Thus, we consider $x \in \mathbb{R}_+$.
- We've constructed the aforementioned s, t graph G as previously mentioned, where for each $e \in E$, we've got a node in G, along with additional nodes (and edges) s, t.
- We maintaine $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i} \leq x$ and thus $y \in P_{\text{ind. set}}$.
- From this, we can obtain the following theorem (most violated inequality, then, is given by $\{e \in E : x(e) > y(e)\}$).

Augmenting path theorem

- Thus, we consider $x \in \mathbb{R}_+$.
- We've constructed the aforementioned s, t graph G as previously mentioned, where for each $e \in E$, we've got a node in G, along with additional nodes (and edges) s, t.
- We maintaine $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i} \leq x$ and thus $y \in P_{\text{ind. set}}$.
- From this, we can obtain the following theorem (most violated inequality, then, is given by $\{e \in E : x(e) > y(e)\}$).

Theorem 4.1

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

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

1. First, assume that there is no such s, t path in G.

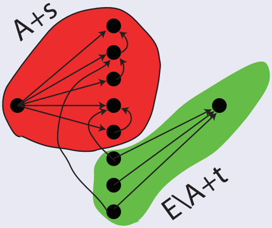
Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

1. First, assume that there is no such s, t path in G.
2. Choose the set $A \subseteq E$ such that **no** edge of G leaves the set $\{s\} \cup A$. That is, choose A minimally corresponding to the zero-edge cut $(A + s, E \setminus A + t)$.



Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

1. First, assume that there is no such s, t path in G.
2. Choose the set $A \subseteq E$ such that **no** edge of G leaves the set $\{s\} \cup A$. That is, choose A minimally corresponding to the zero-edge cut $(A + s, E \setminus A + t)$.
3. Consider $E \setminus A$, and consider any $e \in E \setminus A$, then we must have $y_e = x_e$, or otherwise (if $y_e < x_e$) we'd have an edge (s, e) leaving s.

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

1. First, assume that there is no such s, t path in G.
2. Choose the set $A \subseteq E$ such that **no** edge of G leaves the set $\{s\} \cup A$. That is, choose A minimally corresponding to the zero-edge cut $(A + s, E \setminus A + t)$.
3. Consider $E \setminus A$, and consider any $e \in E \setminus A$, then we must have $y_e = x_e$, or otherwise (if $y_e < x_e$) we'd have an edge (s, e) leaving s.
4. Therefore, $y(E \setminus A) = x(E \setminus A)$.

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

5. Now, consider A. Thus, for all $e \in A$, there is an edge (s, e) and $y_e < x_e$ (by graph construction).

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

5. Now, consider A. Thus, for all $e \in A$, there is an edge (s, e) and $y_e < x_e$ (by graph construction).
6. For all $i \in J$, we claim that $|I_i \cap A| = r(A)$ (which means that for all $i \in J$ and all $e \in A \setminus I_i$, $(I_i \cap A) + e \notin \mathcal{I}$).

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

5. Now, consider A. Thus, for all $e \in A$, there is an edge (s, e) and $y_e < x_e$ (by graph construction).
6. For all $i \in J$, we claim that $|I_i \cap A| = r(A)$ (which means that for all $i \in J$ and all $e \in A \setminus I_i$, $(I_i \cap A) + e \notin \mathcal{I}$).
7. If (6) is not true, then $\exists i \in J$ and $e \in A \setminus I_i$ s.t. $(I_i \cap A) + e \in \mathcal{I}$.

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

5. Now, consider A. Thus, for all $e \in A$, there is an edge (s, e) and $y_e < x_e$ (by graph construction).
6. For all $i \in J$, we claim that $|I_i \cap A| = r(A)$ (which means that for all $i \in J$ and all $e \in A \setminus I_i$, $(I_i \cap A) + e \notin \mathcal{I}$).
7. If (6) is not true, then $\exists i \in J$ and $e \in A \setminus I_i$ s.t. $(I_i \cap A) + e \in \mathcal{I}$.
8. There are two way for (7) to happen ((9) or (10)). That is, either:

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

5. Now, consider A. Thus, for all $e \in A$, there is an edge (s, e) and $y_e < x_e$ (by graph construction).
6. For all $i \in J$, we claim that $|I_i \cap A| = r(A)$ (which means that for all $i \in J$ and all $e \in A \setminus I_i$, $(I_i \cap A) + e \notin \mathcal{I}$).
7. If (6) is not true, then $\exists i \in J$ and $e \in A \setminus I_i$ s.t. $(I_i \cap A) + e \in \mathcal{I}$.
8. There are two way for (7) to happen ((9) or (10)). That is, either:
9. we have that $I_i + e \in \mathcal{I}$, implying that (e, t) is an edge in G (impossible since $(s, e) \in G$, so can't also have $(e, t) \in G$ since no s, t path in G).

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

10. alternatively, $I_i + e \notin \mathcal{I}$, so circuit $C(I_i, e)$ exists which can not be contained in A. *(we needed in (7) that $(I_i \cap A) + e$ is independent, and if the circuit was fully in A then this independence consequence would not hold).*

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

10. alternatively, $I_i + e \notin \mathcal{I}$, so circuit $C(I_i, e)$ exists which can not be contained in A.
11. Thus, in this case, there exists $f \in I_i \setminus A$ such that $f \in C(I_i, e)$, and also edge $(e, f) \in G$. But this also can't happen since this would be a zero-edge cut crossing edge. Thus, (6) is true.

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

10. alternatively, $I_i + e \notin \mathcal{I}$, so circuit $C(I_i, e)$ exists which can not be contained in A.
11. Thus, in this case, there exists $f \in I_i \setminus A$ such that $f \in C(I_i, e)$, and also edge $(e, f) \in G$. But this also can't happen since this would be a zero-edge cut crossing edge. Thus, (6) is true.
12. Therefore, since $y = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}$, we have:

$$y(A) = \sum_{a \in A} y_a = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i}(A) \tag{6}$$

$$= \sum_{i \in J} \lambda_i |I_i \cap A| = \sum_{i \in J} \lambda_i r(A) = r(A) \qquad \text{as required.} \tag{7}$$

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

13. Conversely, suppose there is a directed s, t path in G, and it is given by ordered sequence $S = (e_0, e_1, e_2, \ldots, e_m, e_{m+1})$ of distinct elements, with $e_0 = s$, $e_{m+1} = t$, and $e_i \in E$ for $1 \leq i \leq m$.

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

13. Conversely, suppose there is a directed s, t path in G, and it is given by ordered sequence $S = (e_0, e_1, e_2, \ldots, e_m, e_{m+1})$ of distinct elements, with $e_0 = s$, $e_{m+1} = t$, and $e_i \in E$ for $1 \leq i \leq m$.
14. Then there is a mapping $i_S : [m] \to J$ (where $i_S(j)$ not necessarily distinct, i.e., could have $i_S(j) = i_S(j')$ for $j \neq j'$), satisfying

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

13. Conversely, suppose there is a directed s, t path in G, and it is given by ordered sequence $S = (e_0, e_1, e_2, \ldots, e_m, e_{m+1})$ of distinct elements, with $e_0 = s$, $e_{m+1} = t$, and $e_i \in E$ for $1 \leq i \leq m$.
14. Then there is a mapping $i_S : [m] \to J$ (where $i_S(j)$ not necessarily distinct, i.e., could have $i_S(j) = i_S(j')$ for $j \neq j'$), satisfying

$$(s, e_1) \in G \Rightarrow x_{e_1} > y_{e_1}; \tag{8}$$

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

13. Conversely, suppose there is a directed s, t path in G, and it is given by ordered sequence $S = (e_0, e_1, e_2, \ldots, e_m, e_{m+1})$ of distinct elements, with $e_0 = s$, $e_{m+1} = t$, and $e_i \in E$ for $1 \leq i \leq m$.
14. Then there is a mapping $i_S : [m] \to J$ (where $i_S(j)$ not necessarily distinct, i.e., could have $i_S(j) = i_S(j')$ for $j \neq j'$), satisfying

$$(s, e_1) \in G \Rightarrow x_{e_1} > y_{e_1}; \tag{8}$$

$$(e_1, e_2) \in G \Rightarrow e_2 \in C(I_{i_S(1)}, e_1); \tag{9}$$

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

13. Conversely, suppose there is a directed s, t path in G, and it is given by ordered sequence $S = (e_0, e_1, e_2, \dots, e_m, e_{m+1})$ of distinct elements, with $e_0 = s$, $e_{m+1} = t$, and $e_i \in E$ for $1 \leq i \leq m$.
14. Then there is a mapping $i_S : [m] \to J$ (where $i_S(j)$ not necessarily distinct, i.e., could have $i_S(j) = i_S(j')$ for $j \neq j'$), satisfying

$$(s, e_1) \in G \Rightarrow x_{e_1} > y_{e_1}; \tag{8}$$

$$(e_1, e_2) \in G \Rightarrow e_2 \in C(I_{i_S(1)}, e_1); \tag{9}$$

$$(e_j, e_{j+1}) \in G \Rightarrow e_{j+1} \in C(I_{i_S(j)}, e_j) \text{ for } 1 \leq j \leq m-1; \tag{10}$$

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

13. Conversely, suppose there is a directed s, t path in G, and it is given by ordered sequence $S = (e_0, e_1, e_2, \dots, e_m, e_{m+1})$ of distinct elements, with $e_0 = s$, $e_{m+1} = t$, and $e_i \in E$ for $1 \leq i \leq m$.
14. Then there is a mapping $i_S : [m] \to J$ (where $i_S(j)$ not necessarily distinct, i.e., could have $i_S(j) = i_S(j')$ for $j \neq j'$), satisfying

$$(s, e_1) \in G \Rightarrow x_{e_1} > y_{e_1}; \tag{8}$$

$$(e_1, e_2) \in G \Rightarrow e_2 \in C(I_{i_S(1)}, e_1); \tag{9}$$

$$(e_j, e_{j+1}) \in G \Rightarrow e_{j+1} \in C(I_{i_S(j)}, e_j) \text{ for } 1 \leq j \leq m-1; \tag{10}$$

$$(e_m, t) \in G \Rightarrow I_{i_S(m)} + e_m \in \mathcal{I}. \tag{11}$$

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

15. Now, for $i \in J$, define $k_i = |\{j : i = i_S(j)\}| = \cup_{j \in J} 1_{i = i_S(j)}$ be the number of times that the i'th independent set I_i is used in the mapping $i : [m] \to J$.

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

15. Now, for $i \in J$, define $k_i = |\{j : i = i_S(j)\}| = \cup_{j \in J} 1_{i = i_S(j)}$ be the number of times that the i'th independent set I_i is used in the mapping $i : [m] \to J$.
16. Let $\delta > 0$ be small.

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

15. Now, for $i \in J$, define $k_i = |\{j : i = i_S(j)\}| = \cup_{j \in J} 1_{i = i_S(j)}$ be the number of times that the i'th independent set I_i is used in the mapping $i : [m] \to J$.
16. Let $\delta > 0$ be small.
17. Update λ: For each $i \in J$, set $\lambda_i \leftarrow \lambda_i - k_i \delta$. So now, $\sum_i \lambda_i = 1 - m\delta$ since $\sum_i k_i = m$, leaving the λ_i's deficient.

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

15. Now, for $i \in J$, define $k_i = |\{j : i = i_S(j)\}| = \cup_{j \in J} 1_{i = i_S(j)}$ be the number of times that the i'th independent set I_i is used in the mapping $i : [m] \to J$.
16. Let $\delta > 0$ be small.
17. Update λ: For each $i \in J$, set $\lambda_i \leftarrow \lambda_i - k_i \delta$. So now, $\sum_i \lambda_i = 1 - m\delta$ since $\sum_i k_i = m$, leaving the λ_i's deficient.
18. Next, we add e_1 to $\cup_i I_i$, and distribute amongst them to remove any circuits, as follows.

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

for $j \in 1 \dots m-1$ **do**

  $I_{i_S(j)} \leftarrow I_{i_S(j)} + e_j - e_{j+1}$ and $\lambda_{i_S(j)} \leftarrow \lambda_{i_S(j)} + \delta$;

$I_{i_S(m)} \leftarrow I_{i_S(m)} + e_m$; and $\lambda_{i_S(m)} \leftarrow \lambda_{i_S(m)} + \delta$;

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

for $j \in 1 \dots m-1$ **do**
    $I_{i_S(j)} \leftarrow I_{i_S(j)} + e_j - e_{j+1}$ and $\lambda_{i_S(j)} \leftarrow \lambda_{i_S(j)} + \delta$;
$I_{i_S(m)} \leftarrow I_{i_S(m)} + e_m$; and $\lambda_{i_S(m)} \leftarrow \lambda_{i_S(m)} + \delta$;

19. Due to how G is constructed, the updated I_i are all still independent (we add e_j to $I_{i_S(j)}$ creating a circuit broken by removing e_{j+1}).

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

for $j \in 1 \ldots m-1$ **do**
  $I_{i_S(j)} \leftarrow I_{i_S(j)} + e_j - e_{j+1}$ and $\lambda_{i_S(j)} \leftarrow \lambda_{i_S(j)} + \delta$;
$I_{i_S(m)} \leftarrow I_{i_S(m)} + e_m$; and $\lambda_{i_S(m)} \leftarrow \lambda_{i_S(m)} + \delta$;

19. Due to how G is constructed, the updated I_i are all still independent (we add e_j to $I_{i_S(j)}$ creating a circuit broken by removing e_{j+1}).
20. Also, we again have $\sum_i \lambda_i = 1$. Choose $\delta > 0$ small enough for $\lambda_i \geq 0$, and that $y' = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i} = y + \delta \mathbf{1}_{e_1} \in P$.

Augmenting path theorem

Theorem 4.1

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

Proof of Thm 4.1.

for $j \in 1 \dots m-1$ **do**
  $I_{i_S(j)} \leftarrow I_{i_S(j)} + e_j - e_{j+1}$ and $\lambda_{i_S(j)} \leftarrow \lambda_{i_S(j)} + \delta$;
$I_{i_S(m)} \leftarrow I_{i_S(m)} + e_m$; and $\lambda_{i_S(m)} \leftarrow \lambda_{i_S(m)} + \delta$;

19. Due to how G is constructed, the updated I_i are all still independent (we add e_j to $I_{i_S(j)}$ creating a circuit broken by removing e_{j+1}).
20. Also, we again have $\sum_i \lambda_i = 1$. Choose $\delta > 0$ small enough for $\lambda_i \geq 0$, and that $y' = \sum_{i \in J} \lambda_i \mathbf{1}_{I_i} = y + \delta \mathbf{1}_{e_1} \in P$.
21. The theorem is proven.

Augmenting path theorem consequences

Corollary 4.2

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

$$\max\left(y(E) : y \leq x, y \in P_r\right) = \min\left(x(A) + r(E \setminus A) : A \subseteq E\right) \tag{12}$$

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

Proof.

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

$$y(E) = y(A) + y(E \setminus A) \leq r(A) + x(E \setminus A) \tag{13}$$

as we have seen.

Augmenting path theorem consequences

Corollary 4.2

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

$$\max\left(y(E) : y \leq x, y \in P_r\right) = \min\left(x(A) + r(E \setminus A) : A \subseteq E\right) \qquad (12)$$

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

Proof.

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

$$y(E) = y(A) + y(E \setminus A) \leq r(A) + x(E \setminus A) \qquad (13)$$

   as we have seen.

2. So we need only find a y giving equality.

Augmenting path theorem consequences

Corollary 4.2

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

$$\max\left(y(E) : y \leq x, y \in P_r\right) = \min\left(x(A) + r(E \setminus A) : A \subseteq E\right) \quad (12)$$

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

Proof.

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

$$y(E) = y(A) + y(E \setminus A) \leq r(A) + x(E \setminus A) \quad (13)$$

   as we have seen.

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

Augmenting path theorem consequences

Corollary 4.2

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

$$\max\left(y(E) : y \leq x, y \in P_r\right) = \min\left(x(A) + r(E \setminus A) : A \subseteq E\right) \tag{12}$$

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

Proof.

1. First, any $y \in P$ with $y \leq x$, and any $A \subset E$, we have
$$y(E) = y(A) + y(E \setminus A) \leq r(A) + x(E \setminus A) \tag{13}$$
as we have seen.
2. So we need only find a y giving equality.
3. Choose any $y \in P$ such that $y \leq x$ and with $y(E)$ maximum.
4. 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 4.2

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

$$\max\left(y(E) : y \leq x, y \in P_r\right) = \min\left(x(A) + r(E \setminus A) : A \subseteq E\right) \tag{12}$$

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

Proof.

1. First, any $y \in P$ with $y \leq x$, and any $A \subset E$, we have
$$y(E) = y(A) + y(E \setminus A) \leq r(A) + x(E \setminus A) \tag{13}$$
as we have seen.
2. So we need only find a y giving equality.
3. Choose any $y \in P$ such that $y \leq x$ and with $y(E)$ maximum.
4. 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).
5. 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.

□

Augmenting path theorem consequences

Corollary 4.3

Given matroid M, we have

$$P_{ind.\ set} = P_r \tag{14}$$

We even get this a consequence!

Proof.

- We saw before (in lecture 7) that this follows from corollary 4.2 (which we encountered in lecture 7).

□

Augmenting path theorem consequences

Corollary 4.3

Given matroid M, we have

$$P_{ind.\ set} = P_r \tag{14}$$

We even get this a consequence!

Proof.

- We saw before (in lecture 7) that this follows from corollary 4.2 (which we encountered in lecture 7).
- Therefore, the equivalence follows indirectly just from Theorem 4.1!!

□

Augmenting paths

- The above theorem clearly defines an algorithm, but the question is how efficient it is.

Augmenting paths

- The above theorem clearly defines an algorithm, but the question is how efficient it is.
- An analysis similar to the augmenting-path analysis of Edmonds/Karp for max-flow can be used here as well, and we'll outline these steps (details in Cunningham-84).

Augmenting paths

- The above theorem clearly defines an algorithm, but the question is how efficient it is.
- An analysis similar to the augmenting-path analysis of Edmonds/Karp for max-flow can be used here as well, and we'll outline these steps (details in Cunningham-84).
- Key in the max-flow result (to achieve polynomial time) is to always use the shortest path possible (so always use shortcut free paths, ones for which no shortcut exists).

Augmenting paths

- The above theorem clearly defines an algorithm, but the question is how efficient it is.
- An analysis similar to the augmenting-path analysis of Edmonds/Karp for max-flow can be used here as well, and we'll outline these steps (details in Cunningham-84).
- Key in the max-flow result (to achieve polynomial time) is to always use the shortest path possible (so always use shortcut free paths, ones for which no shortcut exists).
- To find a short-cut free path, we "scan" and "label" nodes, where a node is scanned by examining all incident edges, and labels are given to any previously unlabeled adjacent nodes.

Augmenting paths

- The above theorem clearly defines an algorithm, but the question is how efficient it is.
- An analysis similar to the augmenting-path analysis of Edmonds/Karp for max-flow can be used here as well, and we'll outline these steps (details in Cunningham-84).
- Key in the max-flow result (to achieve polynomial time) is to always use the shortest path possible (so always use shortcut free paths, ones for which no shortcut exists).
- To find a short-cut free path, we "scan" and "label" nodes, where a node is scanned by examining all incident edges, and labels are given to any previously unlabeled adjacent nodes.
- Key in this is to: 1) scan nodes in the order that they are labeled, and 2) label nodes (from a node being scanned) in an order consistent with some fixed total order on all vertices.

Augmenting paths

- The above theorem clearly defines an algorithm, but the question is how efficient it is.
- An analysis similar to the augmenting-path analysis of Edmonds/Karp for max-flow can be used here as well, and we'll outline these steps (details in Cunningham-84).
- Key in the max-flow result (to achieve polynomial time) is to always use the shortest path possible (so always use shortcut free paths, ones for which no shortcut exists).
- To find a short-cut free path, we "scan" and "label" nodes, where a node is scanned by examining all incident edges, and labels are given to any previously unlabeled adjacent nodes.
- Key in this is to: 1) scan nodes in the order that they are labeled, and 2) label nodes (from a node being scanned) in an order consistent with some fixed total order on all vertices.
- While 1) ensures that the path has as few edges as possible (proven in Edmonds/Karp), 2) results in a lexicographically minimum order. Both together are called a *consistent breadth-first search*, or CBFS.

Augmenting paths

- The two requirements for efficiency on an augmenting path based on CBFS are:

Augmenting paths

- The two requirements for efficiency on an augmenting path based on CBFS are:
  1. In every path used for augmentation, there is one critical edge, that becomes unavailable for use immediately after the augmentation.

Augmenting paths

- The two requirements for efficiency on an augmenting path based on CBFS are:
  1. In every path used for augmentation, there is one critical edge, that becomes unavailable for use immediately after the augmentation.
  2. The only way an edge becomes available for use in an augmenting path is by being used in the opposite direction in the previous augmentation.

Augmenting paths

- The two requirements for efficiency on an augmenting path based on CBFS are:
  1. In every path used for augmentation, there is one critical edge, that becomes unavailable for use immediately after the augmentation.
  2. The only way an edge becomes available for use in an augmenting path is by being used in the opposite direction in the previous augmentation.
- On our current context, we have results quite similar to this that guarantee that the number of augmentations is polynomially bounded, yielding our next theorem.

Bounding the number of augmenting paths

- Consider the algorithm implied by Theorem 4.1 as producing one augmentation, and let G_i refer to the digraph at outer iteration i. Then we have

Bounding the number of augmenting paths

- Consider the algorithm implied by Theorem 4.1 as producing one augmentation, and let G_i refer to the digraph at outer iteration i. Then we have

Theorem 4.4

Let $G_0, G_1, \ldots, G_k$ be a sequence of digraphs, each having vertex set $E \cup \{s, t\}$, and correspond to such graphs each one running the algorithm implied by theorem 4.1 Assume fixed total order of $E \cup \{s\}$. Let Q_i denote the CBFS path in G_i, for $0 \leq i < k$. If it is the case that, for $0 \leq i < k$:

Bounding the number of augmenting paths

- Consider the algorithm implied by Theorem 4.1 as producing one augmentation, and let G_i refer to the digraph at outer iteration i. Then we have

Theorem 4.4

Let $G_0, G_1, \ldots, G_k$ be a sequence of digraphs, each having vertex set $E \cup \{s,t\}$, and correspond to such graphs each one running the algorithm implied by theorem 4.1 Assume fixed total order of $E \cup \{s\}$. Let Q_i denote the CBFS path in G_i, for $0 \le i < k$. If it is the case that, for $0 \le i < k$:

Bounding the number of augmenting paths

- Consider the algorithm implied by Theorem 4.1 as producing one augmentation, and let G_i refer to the digraph at outer iteration i. Then we have

Theorem 4.4

Let $G_0, G_1, \ldots, G_k$ be a sequence of digraphs, each having vertex set $E \cup \{s, t\}$, and correspond to such graphs each one running the algorithm implied by theorem 4.1 Assume fixed total order of $E \cup \{s\}$. Let Q_i denote the CBFS path in G_i, for $0 \leq i < k$. If it is the case that, for $0 \leq i < k$:

1. *There is an edge in Q_i that is not an edge in G_{i+1},*

Bounding the number of augmenting paths

- Consider the algorithm implied by Theorem 4.1 as producing one augmentation, and let G_i refer to the digraph at outer iteration i. Then we have

Theorem 4.4

Let $G_0, G_1, \ldots, G_k$ be a sequence of digraphs, each having vertex set $E \cup \{s, t\}$, and correspond to such graphs each one running the algorithm implied by theorem 4.1 Assume fixed total order of $E \cup \{s\}$. Let Q_i denote the CBFS path in G_i, for $0 \leq i < k$. If it is the case that, for $0 \leq i < k$:

1. *There is an edge in Q_i that is not an edge in G_{i+1},*
2. *If (e, f) is an edge in G_{i+1} but not in G_i, then $e, f \in E$ and there are vertices $a, b \in Q_i$ with a preceding b on Q_i such that: 1) either $a = f$ or (a, f) is an edge in G_i; and 2) $b = e$ or (e, b) is an edge in G_i,*

Bounding the number of augmenting paths

- Consider the algorithm implied by Theorem 4.1 as producing one augmentation, and let G_i refer to the digraph at outer iteration i. Then we have

Theorem 4.4

Let $G_0, G_1, \dots, G_k$ be a sequence of digraphs, each having vertex set $E \cup \{s, t\}$, and correspond to such graphs each one running the algorithm implied by theorem 4.1 Assume fixed total order of $E \cup \{s\}$. Let Q_i denote the CBFS path in G_i, for $0 \leq i < k$. If it is the case that, for $0 \leq i < k$:

1. *There is an edge in Q_i that is not an edge in G_{i+1},*
2. *If (e, f) is an edge in G_{i+1} but not in G_i, then $e, f \in E$ and there are vertices $a, b \in Q_i$ with a preceding b on Q_i such that: 1) either $a = f$ or (a, f) is an edge in G_i; and 2) $b = e$ or (e, b) is an edge in G_i,*

Then we have that the number of augmentations has bound $k \leq |E|^3$.

Bounding the number of augmenting paths

Theorem 4.5

It is possible to construct an augmentation scheme such that each augmenting path is done in accordance to Theorem 4.4. Each such augmentation is CBFS, and is called a "grand" augmentation, and is maximal in a certain way. This achieves the $O(n^3)$ time, in the number of augmentations, mentioned above.

Bounding the number of augmenting paths

Theorem 4.5

It is possible to construct an augmentation scheme such that each augmenting path is done in accordance to Theorem 4.4. Each such augmentation is CBFS, and is called a "grand" augmentation, and is maximal in a certain way. This achieves the $O(n^3)$ time, in the number of augmentations, mentioned above.

- Of course, the cost of each augmentation might be expensive. For matric matroids, each would be $O(r^2 n^5)$ where r is the number of rows of the matrix, leading to $O(r^2 n^8)$ algorithm.

Bounding the number of augmenting paths

Theorem 4.5

It is possible to construct an augmentation scheme such that each augmenting path is done in accordance to Theorem 4.4. Each such augmentation is CBFS, and is called a "grand" augmentation, and is maximal in a certain way. This achieves the $O(n^3)$ time, in the number of augmentations, mentioned above.

- Of course, the cost of each augmentation might be expensive. For matric matroids, each would be $O(r^2 n^5)$ where r is the number of rows of the matrix, leading to $O(r^2 n^8)$ algorithm.
- On the other hand, this algorithm has some intriguing properties.

Towards SFM

- Recall the Edmonds matroid partition algorithm, was SFM for $r(A) - \frac{1}{k}\mathbf{1}(A)$.

Towards SFM

- Recall the Edmonds matroid partition algorithm, was SFM for $r(A) - \frac{1}{k}\mathbf{1}(A)$.
- We now have an algorithm that can do $r(A) - x(A)$ for any $x \in \mathbb{R}_+^E$.

Towards SFM

- Recall the Edmonds matroid partition algorithm, was SFM for $r(A) - \frac{1}{k}\mathbf{1}(A)$.
- We now have an algorithm that can do $r(A) - x(A)$ for any $x \in \mathbb{R}_+^E$.
- There are three limitations to this:

Towards SFM

- Recall the Edmonds matroid partition algorithm, was SFM for $r(A) - \frac{1}{k}\mathbf{1}(A)$.
- We now have an algorithm that can do $r(A) - x(A)$ for any $x \in \mathbb{R}_+^E$.
- There are three limitations to this:
  1. $r(A)$ is only a matroid rank function (and thus integral) rather than a (possibly non-integral) polymatroidal function.

Towards SFM

- Recall the Edmonds matroid partition algorithm, was SFM for $r(A) - \frac{1}{k}\mathbf{1}(A)$.
- We now have an algorithm that can do $r(A) - x(A)$ for any $x \in \mathbb{R}_+^E$.
- There are three limitations to this:
    1. $r(A)$ is only a matroid rank function (and thus integral) rather than a (possibly non-integral) polymatroidal function.
    2. x is required to be positive $x \geq 0$.

Towards SFM

- Recall the Edmonds matroid partition algorithm, was SFM for $r(A) - \frac{1}{k}\mathbf{1}(A)$.
- We now have an algorithm that can do $r(A) - x(A)$ for any $x \in \mathbb{R}_+^E$.
- There are three limitations to this:
  1. $r(A)$ is only a matroid rank function (and thus integral) rather than a (possibly non-integral) polymatroidal function.
  2. x is required to be positive $x \geq 0$.
  3. This works only for the difference between r and x, but we'd like an algorithm that works for any arbitrary submodular function f, even non-monotone and/or non-non-increasing/decreasing.

Towards SFM

- Recall the Edmonds matroid partition algorithm, was SFM for $r(A) - \frac{1}{k}\mathbf{1}(A)$.
- We now have an algorithm that can do $r(A) - x(A)$ for any $x \in \mathbb{R}_+^E$.
- There are three limitations to this:
  1. $r(A)$ is only a matroid rank function (and thus integral) rather than a (possibly non-integral) polymatroidal function.
  2. x is required to be positive $x \geq 0$.
  3. This works only for the difference between r and x, but we'd like an algorithm that works for any arbitrary submodular function f, even non-monotone and/or non-non-increasing/decreasing.
- It turns out that (2) and (3) is easy to deal with, but (1) took another 16 years to solve (and perhaps can still be seen as unsolved, w.r.t. wanting a scalable algorithm).

Towards SFM

- First, given any submodular function g, construct modular function $m : E \to \mathbb{R}$ such that $m(e) = g(E \setminus \{e\}) - g(E)$.

Towards SFM

- First, given any submodular function g, construct modular function $m : E \to \mathbb{R}$ such that $m(e) = g(E \setminus \{e\}) - g(E)$.
- Then construct a new function $f : 2^E \mathbb{R}_+$ as

$$f(A) = g(A) + m(A) - g(\emptyset) \tag{15}$$

Towards SFM

- First, given any submodular function g, construct modular function $m : E \to \mathbb{R}$ such that $m(e) = g(E \setminus \{e\}) - g(E)$.
- Then construct a new function $f : 2^E \mathbb{R}_+$ as

$$f(A) = g(A) + m(A) - g(\emptyset) \tag{15}$$

- Then $f(\emptyset) = 0$, so f is normalized.

Towards SFM

- First, given any submodular function g, construct modular function $m : E \to \mathbb{R}$ such that $m(e) = g(E \setminus \{e\}) - g(E)$.
- Then construct a new function $f : 2^E \mathbb{R}_+$ as

$$f(A) = g(A) + m(A) - g(\emptyset) \tag{15}$$

- Then $f(\emptyset) = 0$, so f is normalized.
- Also, f is monotone non-decreasing and submodular. It is submodular since sum of submodular and modular. Monotone non-decreasing follows since

$$\begin{aligned} f(B+v) - f(B) &= g(B+v) - g(B) + m(v) && (16)\\ &= g(B+v) - g(B) + g(E-v) - g(E) && (17)\\ &\geq 0 && (18) \end{aligned}$$

since, by submodularity, $g(B+v) - g(B) \geq -g(E-v) + g(E)$.

Towards SFM

- Also, given $f(A) = g(A) + m(A) - g(\emptyset)$, to minimize g we can just minimize $f - m$.

Towards SFM

- Also, given $f(A) = g(A) + m(A) - g(\emptyset)$, to minimize g we can just minimize $f - m$.
- So now we have a difference of a polymatroid function and a modular function. This deals with (3) above.

Towards SFM

- Also, given $f(A) = g(A) + m(A) - g(\emptyset)$, to minimize g we can just minimize $f - m$.
- So now we have a difference of a polymatroid function and a modular function. This deals with (3) above.
- Is $m \in \mathbb{R}_+^E$?

Towards SFM

- Also, given $f(A) = g(A) + m(A) - g(\emptyset)$, to minimize g we can just minimize $f - m$.
- So now we have a difference of a polymatroid function and a modular function. This deals with (3) above.
- Is $m \in \mathbb{R}_+^E$?
- No, but for any e such that $m(e) < 0$ can't be a minimizer of $f - m$ since, assuming that A minimizes $f(A) - m(A)$ and $e \in A$ is such that $m(e) < 0$, then we have that $f(A') - m(A') < f(A) - m(A)$ where $A' = A \setminus \{e\}$.

Towards SFM

- Also, given $f(A) = g(A) + m(A) - g(\emptyset)$, to minimize g we can just minimize $f - m$.
- So now we have a difference of a polymatroid function and a modular function. This deals with (3) above.
- Is $m \in \mathbb{R}_+^E$?
- No, but for any e such that $m(e) < 0$ can't be a minimizer of $f - m$ since, assuming that A minimizes $f(A) - m(A)$ and $e \in A$ is such that $m(e) < 0$, then we have that $f(A') - m(A') < f(A) - m(A)$ where $A' = A \setminus \{e\}$.
- This follows since f is monotone non-decreasing, and $m(A) = m(A') + m(e)$. This deals with (2) above.

Towards SFM

- Also, given $f(A) = g(A) + m(A) - g(\emptyset)$, to minimize g we can just minimize $f - m$.
- So now we have a difference of a polymatroid function and a modular function. This deals with (3) above.
- Is $m \in \mathbb{R}_+^E$?
- No, but for any e such that $m(e) < 0$ can't be a minimizer of $f - m$ since, assuming that A minimizes $f(A) - m(A)$ and $e \in A$ is such that $m(e) < 0$, then we have that $f(A') - m(A') < f(A) - m(A)$ where $A' = A \setminus \{e\}$.
- This follows since f is monotone non-decreasing, and $m(A) = m(A') + m(e)$. This deals with (2) above.
- Therefore, SFM is as "easy" as moving from matroid rank functions to not-necessarily-integral polymatroidal functions.

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.