CMPUT 675: Combinatorial Optimization	**Winter 2021**

Lecture 20 (March 24): Matroid Intersection

Lecturer: Zachary Friggstad *Scribe: Saeed Najafi*

20.1 More Applications of Matroid Intersection

We provide the following problems that can be solved with Matroid Intersection. Let $M_1 = (X, I_1)$ and $M_2 = (X, I_2)$ be the two matroids defined over the same set of items X. We are looking for the the maximal (largest) $A \subseteq X$ such that $A \in I_1 \cap I_2$.

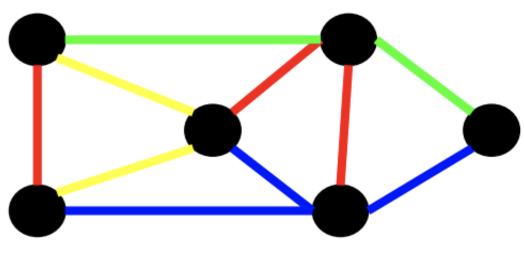

Figure 20.1: An example of a colorful spanning tree where we can only select 2 red, 1 blue, 2 yellow and 2 green edges.

- Example 1) Colorful Spanning Trees

  We have the graph $\mathcal{G} = (V, E)$ where each edge has a color. Let $M_1 = (E, I_1)$ be the graphic matroid where $I_1 = \{F \subseteq E : (V; F) \text{ has no cycles}\}$. Define $M_2 = (E, I_2)$ as the partition matroid over the color classes where $I_2 = \{F \subseteq E : \text{number of edges in } F \text{ that have color } c_i \leq \text{the bound on the total number of edges with color } c_i\}$. Figure 20.1 illustrates this example. The $M_1 \cap M_2$ is the set of forests that satisfy the color bounds. We can verify if there is a spanning tree satisfying the color bounds by looking at the largest subset $F \subseteq I_1 \cap I_2$.

- Example 2) Scheduling with Suppliers

  We have a set of unit size jobs J where each has a deadline $d_j \in \mathbb{Z}_{\geq 0}$. Each job must also be assigned to a supplier. The supplier $s \in S$ can supply at most $\mu(s)$ jobs. We can illustrate the problem with a bipartite graph $\mathcal{G} = (J \cup S, E)$ where there is an edge between the job $j \in J$ and the supplier $s \in S$ if j can be supported by s. Figure 20.2 illustrates an example. Let define $M_1 = (J, I_1)$ as the scheduling matroid where $I_1 = \{y \subseteq J : \text{we can process } \forall j \in y \text{ in some order before their deadline}\}$. Let $M_2 = (J, I_2)$ be a generalization of the transversal matroid where $I_2 = \{y \subseteq J : \exists \text{ assignment from the jobs in } y \text{ to the suppliers such that } \delta(s) \leq \mu(s) \text{ for every } s \in S\}$. We can find the largest subset of the jobs that can be scheduled before their deadline while being supported by their suppliers through the matroid intersection $M_1 \cap M_2$.

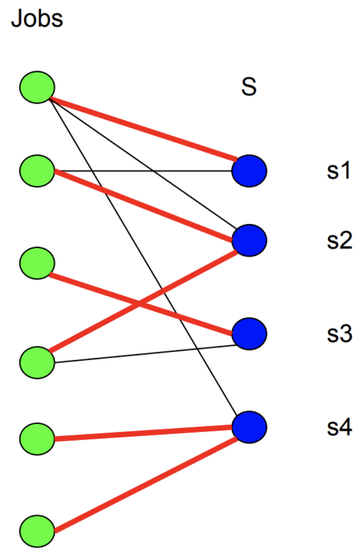

Figure 20.2: An example for the scheduling of the jobs with suppliers. Red lines represent a valid assignment to the suppliers with $\mu(s_1) = 1$, $\mu(s_2) = 4$, $\mu(s_3) = 1$, and $\mu(s_4) = 2$.

- Example 3) Arborescence matroid

  The ground set $X = E$ is set of the directed edges in a directed graph $\mathcal{G} = (V, E)$. The M_1 is the graphic matroid ignoring the directions of the edges. Let $M_2 = (X, I_2)$ be a matroid defined over the directed edges where $I_2 = \{y \subseteq X : \delta_y^{in}(v) \leq 1$ for every node $v \in V - \{root\}\}$. In this directed graph, an arborescence T is $T \in I_1 \cap I_2$ such that $|T| = |V| - 1$.

20.2 Weighted Matroid Intersection

Let $M_1 = (X, I_1)$ and $M_2 = (X, I_2)$ be the two matroids defined over the same set of items X. We also have non-negative weights $w : X \to \mathbb{R}_{\geq 0}$ over the items X. Let define the variable x_i for the item $i \in X$. We can use the following LP to find the weighted intersection of these two matroids where r_1 and r_2 are the rank functions corresponding to M_1 and M_2, respectively.

$$\begin{array}{rrcll} \textbf{maximize}: & \sum_{i \in X} w(i) \cdot x_i & & & \\ \textbf{subject to}: & x(A) & \leq & r_1(A) & \forall A \subseteq X \\ & x(A) & \leq & r_2(A) & \forall A \subseteq X \\ & x_i & \geq & 0 & \forall i \in X \end{array}$$

Lemma 1 *Given the two constraints defined over only two matroids, extreme points of the previous LP are integral.*

Proof. Let $\overline{x}$ be an extreme point. Without the loss of generality, we can assume $\overline{x}_i > 0$ for all $i \in X$ (restrict the items of the matroids to keep those with $x_i > 0$).

Let $A_1 \subseteq A_2 \subseteq ... \subseteq A_k$ and $B_1 \subseteq B_2 \subseteq ... \subseteq B_l$ be the uncrossing chains for the two matroids (see the previous lecture 19 for the uncrossing lemma). We have the following properties for these chains:

- $\overline{x}(A_j) = r_1(A_j)$ $(1 \le j \le k)$.
- $\overline{x}(B_j) = r_2(B_j)$ $(1 \le j \le l)$.

The $\{\chi_{A_j} : 1 \le j \le k\}$ is a basis for all the tight constraints of the first type $\{\chi_A : \overline{x}(A) = r_1(A)\}$ and $\{\chi_{B_j} : 1 \le j \le l\}$ is a basis for all the tight constraints of the second type $\{\chi_B : \overline{x}(B) = r_2(B)\}$. Hence, $\phi = \{\chi_{A_j} : 1 \le j \le k\} \cup \{\chi_{B_j} : 1 \le j \le l\}$ is a basis for $\mathbb{R}^X_{\ge 0}$. Figure 20.3 illustrates the network matrix corresponding to the constraints in ϕ. For $i \in X$, let $j(i)$ be the minimum index such that $i \in A_{j(i)}$ and let $\overline{j}(i)$ be the minimum index such that $i \in B_{\overline{j}(i)}$ (if $i \notin A_k$, then $j(i) = L$ and if $i \notin B_l$, then $\overline{j}(i) = L$). If we set $s_i = U_{j(i)}$ and $t_i = V_{\overline{j}(i)}$, then the corresponding network matrix is the same as the matrix formed with rows from vectors in ϕ. We can conclude that $\overline{x}$ is integral.

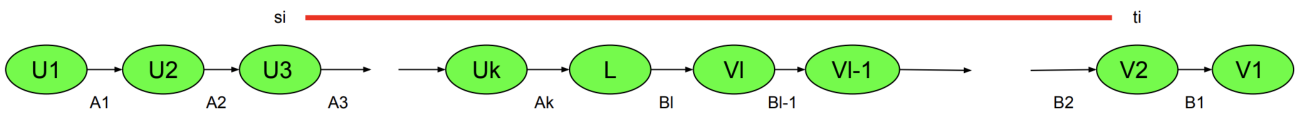

Figure 20.3: The designed network matrix for the rows in ϕ. The edge between U_j and U_{j+1} corresponds to the subset A_j. The edge between V_{j+1} and V_j corresponds to the subset B_j (L denotes U_{k+1} or V_{l+1}).

■

20.3 Combinatorial Algorithm for Matroid Intersection (Max Size)

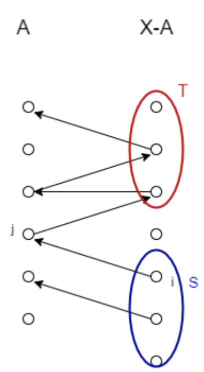

Figure 20.4: The bipartite graph $\mathcal{G}$ constructed using M_1, M_2, and A. There is a directed edge (i, j) from $X - A$ to A if $A + i - j \in I_2$ and there is a directed edge (j, i) from A to $X - A$ if $A + i - j \in I_1$.

Consider the following basic subroutine: Given $A \in I_1 \cap I_2$, find a) $A' \in I_1 \cap I_2$ such that $|A'| = |A| + 1$ or b) declare correctly that A is the max-size set in $I_1 \cap I_2$. Starting from an empty set, we can call this subroutine in a polynomial time to find the set with the maximum size.

Consider the following directed bipartite graph $\mathcal{G} = (A \cup X - A, E)$:

- $S = \{i \in X - A : A + i \in I_1\}$

- $T = \{i \in X - A : A + i \in I_2\}$ (S and T can overlap).
- $E = \{(i,j) : A + i - j \in I_2\} \cup \{(j,i) : A + i - j \in I_1\}$. The edge (i,j) is directed from the right $(X - A)$ to the left (A). The edge (j,i) is directed from the left (A) to the right $(X - A)$. It is clear that we can construct E with at most $O(|X|^2)$ calls to the independence oracle.

Algorithm 1 ALGORITHM FOR MATROID INTERSECTION

Input: $M_1 = (X, I_1)$, $M_2 = (X, I_2)$, and the independence oracle.
Output: The largest A in $I_1 \cap I_2$

$A \leftarrow \emptyset$
$\mathcal{G} = (A \cup X - A, E), S, T \leftarrow$ Construct the described bipartite graph from Figure 20.4 (in time $O(|X|^2)$)
while $\nexists$ path from $S - T$ in $\mathcal{G}$ **do**
  $i_0, j_1, i_1, ..., j_k, i_k \leftarrow$ The shortest path from S to T.
  $A \leftarrow A \cup \{i_0, i_1, i_2, ..., i_k\} - \{j_1, j_2, j_3, ..., j_k\}$.
  $\mathcal{G} = (A \cup X - A, E), S, T \leftarrow$ Construct the described bipartite graph from Figure 20.4 (in time $O(|X|^2)$)
return A

In the rest of this lecture, we will introduce the matroid circuit and two relevant lemmas for circuits. We then prove the feasibility of the solution returned by the mentioned algorithm. The maximality of the algorithm will be proved in the next lecture.

Definition: Let $M = (X, I)$ be a matroid. A circuit is $C \subseteq X$ such that $C \notin I$ but $C - i \in I$ for every $i \in C$. An example would be a simple cycle where the deletion of one edge along the cycle makes the remaining edges to be part of the graphic matroid.

Lemma 2 *Let $Y \in I$ and $i \in X$ such that $Y + i \notin I$. There exists a unique circuit $C(Y, i) \subseteq Y + i$.*

Proof. It is clear that $Y + i$ has a circuit; consider any minimal $C \subseteq Y + i$ such that $C \notin I$. Such C exists and is a circuit.

Lets assume that the opposite is correct and we now have two distinct circuits $C_1, C_2 \subseteq Y + i$. Any proper subset of a circuit is independent, therefore $C_1 \not\subset C_2$ and $C_2 \not\subset C_1$. Since any subset of Y is also independent, we must have $i \in C_1 \cap C_2$ (otherwise $C_1, C_2 \subseteq Y$). Let $j \in C_1 - C_2$. By using the sub-modularity lemma twice, once between C_2 and $C_1 \cup C_2 - \{i,j\}$, and another one between C_1 and $C_1 \cup C_2 - j$, we have:

$$r(C_1) + r(C_2) + r(C_1 \cup C_2 - \{i,j\}) \geq r(C_1) + r(C_2 - i) + r(C_1 \cup C_2 - j) \geq r(C_1 - j) + r(C_2 - i) + r(C_1 \cup C_2).$$

We know that rank of a cycle is the same as the rank of any subset without one item, therefore $r(C_1) = r(C_1 - j)$ and $r(C_2) = r(C_2 - i)$. Therefore, we have $r(C_1 \cup C_2 - \{i,j\}) \geq r(C_1 \cup C_2)$. Since $C_1 \cup C_2 - \{i,j\}$ is a subset of $C_1 \cup C_2$, we also have $r(C_1 \cup C_2) \geq r(C_1 \cup C_2 - \{i,j\})$. Hence, $r(C_1 \cup C_2) = r(C_1 \cup C_2 - \{i,j\})$.

To continue the proof, we know that $C_1 \cup C_2 - \{i,j\} \subseteq C_1 \cup C_2 - i \subseteq C_1 \cup C_2$, so we have $r(C_1 \cup C_2 - i) = r(C_1 \cup C_2 - \{i,j\})$. Since $C_1 \cup C_2 - \{i,j\} \subseteq Y$, then $r(C_1 \cup C_2 - \{i,j\}) = |C_1 \cup C_2 - \{i,j\}|$ (rank of an independent subset, which becomes the maximal independent subset of itself, equals the size of itself).

We now have $r(C_1 \cup C_2 - i) = r(C_1 \cup C_2 - \{i,j\}) = |C_1 \cup C_2 - \{i,j\}| < |C_1 \cup C_2 - i|$, therefore $C_1 \cup C_2 - i \notin I$, but we also have that $C_1 \cup C_2 - i \subseteq Y$ and $C_1 \cup C_2 - i \in I$. This contradiction completes our proof. ∎

Lemma 3 *Let $A \in I$, $i_1, i_2, ..., i_k \in X - A$, and $j_1, j_2, ..., j_k \in A$ such that:*

- *$j_m \in C(A, i_m)$ $(\forall 1 \leq m \leq k)$. This means $A + i_m \notin I$, but adding i_m to A makes j_m to be deleted and still be an independent set.*

- $j_m \notin C(A, i_l)$ ($\forall 1 \leq m < l \leq k$). *No earlier* j_m *lies in the circuit* $C(A, i_l)$.

Then, $A + \{i_1, i_2, ..., i_k\} - \{j_1, j_2, ..., j_k\} \in I$.

Proof. The proof is by induction on k. The base case of $k = 0$ is trivial. Now suppose for $k > 0$, $\overline{A} = A + \{i_1, i_2, ..., i_{k-1}\} - \{j_1, j_2, ..., j_{k-1}\}$ and $\overline{A} \in I$ by induction. We now have the following cases:

- $\overline{A} + i_k \in I$, then $\overline{A} + i_k - j_k$, which is a subset of $\overline{A} + i_k$, is also independent, so the induction claim holds.
- $\overline{A} + i_k \notin I$, then we claim that $C(A, i_k) = C(\overline{A}, i_k)$.

  No j_m with $m < k$ is inside $C(A, i_k)$ according to the assumption. Therefore, $C(A, i_k) \subseteq C(\overline{A}, i_k)$, and by the uniqueness of the circuits, we must have $C(A, i_k) = C(\overline{A}, i_k)$. As we know that $j_k \in C(A, i_k)$, then $j_k \in C(\overline{A}, i_k)$ and we can conclude that $\overline{A} + i_k - j_k \in I$. The induction claim holds in this case as well and we could prove the lemma for all $k \geq 0$.

■

Theorem 1 *Suppose* $C_1(A, i)$ *be the circuit for the first matroid* M_1 *of the algorithm formed by adding* i *to* A *(*$A \in I_1$*, but* $A + i \notin I_1$*). Similarly, Let* $C_2(A, i)$ *be the circuit for the second matroid* M_2 *of the algorithm formed by adding* i *to* A *(*$A \in I_2$*, but* $A + i \notin I_2$*). Let* $i_0, j_1, i_1, ..., j_k, i_k$ *be the shortest* $S - T$ *path in the bipartite graph* $\mathcal{G}$ *of the algorithm. Then,* $A + i_0$*,* $i_1, i_2, ..., i_k$*, and* $j_1, j_2, ..., j_k$ *satisfy the conditions of the lemma 3 for* M_1*. Similarly,* $A + i_k$*,* $i_0, i_1, ..., i_{k-1}$*, and* $j_1, j_2, ..., j_k$ *satisfy the conditions of the lemma 3 for* M_2*.*

This theorem and the result of lemma 3 shows that $A + \{i_0, i_1, ..., i_k\} - \{j_1, j_2, ..., j_k\}$ *is always an independent set in* $I_1 \cap I_2$*.*

Proof of Theorem 1. We prove the first claim that $A + i_0$, $i_1, i_2, ..., i_k$, and $j_1, j_2, ..., j_k$ satisfy the conditions of the lemma 3 for M_1. We know that $A + i_0 \in I_1$ since $i_0 \in S$. Since $i_m \notin S$ for $1 \leq m \leq k$, then $A + i_0 + i_m \notin I_1$ for $1 \leq m \leq k$. Hence, $C_1(A + i_0, i_m) = C_1(A, i_m)$ for $1 \leq m \leq k$. Since we also have the edge (j_m, i_m) in the bipartite graph, then $A + i_m - j_m \in I_1$. We can conclude that $j_m \in C_1(A, i_m) = C_1(A + i_0, i_m)$ for $1 \leq m \leq k$. To verify the second assumption of the lemma 3, we must have $j_m \notin C_1(A + i_0, i_l)$ for $1 \leq m < l \leq k$, otherwise we would have $j_m \in C_1(A, i_l)$ and this means that we would have the shortcut edge (j_m, i_l) in the bipartite graph which cannot be true given the shortest path used in the algorithm.

To prove the second claim of the theorem that $A + i_k$, $i_0, i_2, ..., i_{k-1}$, and $j_1, j_2, ..., j_k$ satisfy the conditions of the lemma 3, we can take a similar argument. We know that $A + i_k \in I_2$ since $i_k \in T$. Since $i_m \notin T$ for $0 \leq m \leq k - 1$, then $A + i_k + i_m \notin I_2$ for $0 \leq m \leq k - 1$. Hence, $C_2(A + i_k, i_m) = C_2(A, i_m)$ for $0 \leq m \leq k - 1$ for M_2. Since we also have the edge (i_{m-1}, j_m) in the bipartite graph, then $A + i_{m-1} - j_m \in I_2$. We can conclude that $j_m \in C_2(A, i_{m-1}) = C_2(A + i_k, i_{m-1})$ for $1 \leq m \leq k$. Furthermore, as we don't have any shortcut edge of (j_m, i_l) for $1 \leq m < l \leq k$ in the shortest path, then $j_m \notin C(A + i_k, i_l)$ for $1 \leq m < l \leq k$. ■

References