

Lecture 2: Matroid Intersection and Bipartite Matchings

Notes by Ola Svensson¹

These lecture notes is a combination of two lecture notes by Michel Goemans:

- <http://www-math.mit.edu/~goemans/18433S09/matroid-intersect-notes.pdf>
- <http://math.mit.edu/~goemans/18433S09/matching-notes.pdf>

1 Recall: Matroids and Greedy

We start by recalling the main points from last lecture: definition of matroids which is exactly the structures for which GREEDY works.

Definition 1 A matroid $\mathcal{M} = (E, \mathcal{I})$ ² is defined on a finite ground set E and a family $\mathcal{I}$ of subsets of E that are called independent sets satisfying two axioms:

(I_1) if $X \subseteq Y$ and $Y \in \mathcal{I}$ then $X \in \mathcal{I}$.

(I_2) if $X \in \mathcal{I}$ and $Y \in \mathcal{I}$ and $|Y| > |X|$ then $\exists e \in Y \setminus X : X \cup \{e\} \in \mathcal{I}$.

Remarks:

- The family $\mathcal{I}$ may be exponential in the size of the ground set E (as is the case for example for graphic matroids of complete graphs). One therefore often assumes that $\mathcal{I}$ is given implicitly by having a membership/independence oracle: an algorithm that given $I \subseteq E$ efficiently answers whether $I \in \mathcal{I}$.
- The second axiom implies that every *maximal* independent set is of maximum cardinality. In other words, all maximal independent sets have the same cardinality. A maximal cardinality set is called a *base* of the matroid.

With the more abstract notation of matroids, the basic greedy algorithm becomes

GREEDY($\mathcal{M}, w$):

Input: A matroid $\mathcal{M} = (E, \mathcal{I})$ and weights $(w_e)_{e \in E}$.

Output: A maximum weight base S .

1. Sort and relabel the elements so that $w_1 \geq w_2 \geq \dots \geq w_{|E|}$.
2. $S \leftarrow \emptyset$.
3. **for** $i = 1$ **to** $|E|$:
4. **if** $S + i \in \mathcal{I}$ **then** $S \leftarrow S + i$.
5. **return** S .

The concept of matroids was defined so as to enable the correctness analysis of the basic greedy algorithm. Perhaps more surprisingly, it is if and only if.

¹**Disclaimer:** These notes were written as notes for the lecturer. They have not been peer-reviewed and may contain inconsistent notation, typos, and omit citations of relevant works.

²In these notes, we use $\mathcal{M}$ (instead of M) for matroids to not confuse them with matchings.

Theorem 2 (Rado'57/Gale'68/Edmonds'71) For any ground set $E = \{1, 2, \dots, n\}$, and a family of subsets $\mathcal{I}$, GREEDY finds a maximum weight base for any set of weights $w : E \rightarrow \mathbb{R}$ if and only if $\mathcal{M} = (E, \mathcal{I})$ is a matroid.

In the last lecture, we saw some examples of matroids (k -Uniform matroids, partition matroids, linear matroids).

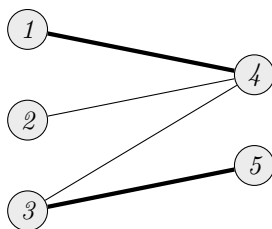
2 Matroid Intersection

Although the concept of matroids forms a rich set of problems that can be solved by the greedy algorithm, there are many problems that have efficient algorithms but are not matroids. Consider for example the bipartite matching problem:

Definition 3 Given a bipartite graph $G = (V, E)$, find a matching of maximum size.

Recall that a matching $M \subseteq E$ is a subset of edges so that every vertex is incident to at most one edge of M , i.e., $|\{e \in M : v \in e\}| \leq 1$ for all $v \in V$.

Example 1 The edges $(1, 4)$ and $(3, 5)$ form a matching (of maximum cardinality).



Let's see if GREEDY works for maximum cardinality (or max weight) bipartite matching. To do this, we can verify if $(E, \mathcal{I})$ is a matroid where

$$\mathcal{I} = \{M \subseteq E : M \text{ is a matching}\}.$$

It is easy to see that $(E, \mathcal{I})$ satisfies the downward closed property (I_1) of matroids since a subset of a matching is still a matching. However, the second axiom (I_2) is not satisfied which can be seen by taking the two matchings $M_1 = \{(1, 4), (3, 5)\}$ and $M_2 = \{(3, 4)\}$: although $|M_1| > |M_2|$ there is no edge in M_1 that we can add to M_2 while maintaining that it is a matching.

This is kind of sad: matroids and simple GREEDY have their limits. On the other hand, we know from our undergraduate algorithm course that there are efficient algorithms for the bipartite matching problem. That there are efficient algorithms for the bipartite matching problem is in fact part of a more general phenomena: it can be described as the intersection of two matroids and any such problem has an efficient algorithm!

Definition 4 Given two matroids $\mathcal{M}_1 = (E, \mathcal{I}_1)$ and $\mathcal{M}_2 = (E, \mathcal{I}_2)$ on the same ground set E , their intersection is

$$\mathcal{M}_1 \cap \mathcal{M}_2 = (E, \mathcal{I}_1 \cap \mathcal{I}_2).$$

Note that the intersection of two matroids satisfy (I_1) but not typically (I_2). The following adds a lot of power to the concepts of matroids:

Theorem 5 (Edmonds, Lawler, 70's) There is an efficient algorithm for finding a max-weight independent set in the intersection of two matroids.³

³It is efficient meaning that it is a polynomial-time algorithm if we can, in polynomial time, answer independence queries of the type " $I \in \mathcal{I}_1$?" and " $I \in \mathcal{I}_2$?" for the two matroids (which is the case for all examples of matroids seen in class).

The algorithm that gives the above result is quite technical and notation heavy. Instead of describing it in its full generality, we first see how we can apply the above result to obtain several interesting algorithmic applications. We then develop an algorithm for the special case of the bipartite matching problem that share several similarities with the general algorithm.

2.1 Examples

This section is basically taken verbatim from the excellent lecture notes by Michel Goemans available here (together with a lot of other interesting information): <http://www-math.mit.edu/~goemans/18433S09/matroid-intersect-notes.pdf>

2.1.1 Bipartite Matching

We already mentioned that the bipartite matching problem is an example of matroid intersection. To see this, consider a bipartite graph $G = (V, E)$ with bipartition (A, B) , i.e., the vertices V are partitioned into two disjoint sets A and B such that every edge is between a vertex in A and a vertex in B .

Let $\mathcal{M}_A$ be a partition matroid with ground set E where the partition E is given by $E = \bigcup \{\delta(v) : v \in A\}$ where $\delta(v)$ denotes the edges incident to v . Notice that this is a partition since all edges have precisely one endpoint in A . We also define $k_v = 1$ for every $v \in A$. Thus, the family of independent sets of $\mathcal{M}_A$ is given by

$$\mathcal{I}_A = \{F \subseteq E : |F \cap \delta(v)| \leq 1 \text{ for all } v \in A\}.$$

In other words, a set of edges is independent for $\mathcal{M}_A$ if it has at most one edge incident to every vertex of A (and any number of edges incident to every vertex of B). We can similarly define $\mathcal{M}_B = (E, \mathcal{I}_B)$ by

$$\mathcal{I}_B = \{F \subseteq E : |F \cap \delta(v)| \leq 1 \text{ for all } v \in B\}.$$

Now observe that any $F \in \mathcal{I}_A \cap \mathcal{I}_B$ corresponds to a matching in G , and vice versa. And the largest common independent set $\mathcal{I}_A$ and $\mathcal{I}_B$ corresponds to a maximum matching in G .

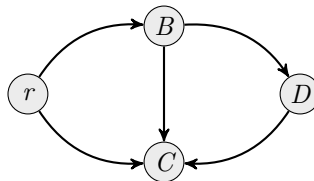
2.1.2 Colorful Spanning Trees

Suppose we have an undirected graph $G = (V, E)$ and every edge has a color. This is represented by a partition of E into $E_1 \cup E_2 \cup \dots \cup E_k$ where each E_i represents a set of edges of the same color i . The problem of deciding whether this graph has a spanning tree in which all edges have a different color can be tackled through matroid intersection. Such a spanning tree is called colorful. Colorful spanning trees are bases of the graphic matroid $\mathcal{M}_1 = (E, \mathcal{I}_1)$ of G which are also independent in the partition matroid $\mathcal{M}_2 = (E, \mathcal{I}_2)$ defined by $\mathcal{I}_2 = \{F \subseteq E : |F \cap E_i| \leq 1 \text{ for all } i\}$.

2.1.3 Arborescences

Given a directed graph $D = (V, A)$ and a special root vertex $r \in V$, an r -arborescence (or just arborescence) is a spanning tree (when viewed as an undirected graph) directed away from r . Thus, in a r -arborescence, every vertex is reachable from the root r . As an r -arborescence has no arc incoming to the root, we assume that D has no such arc.

Example 2 The following directed graph has e.g. the r -arborescence $\{(r, B), (B, C), (B, D)\}$.



r -arborescences can be viewed as sets simultaneously independent in two matroids. Let G denote the undirected counterpart of D obtained by disregarding the directions of the arcs. Note that if we have both arcs $a_1 = (u, v)$ and $a_2 = (v, u)$ in D then we get two undirected edges also labeled a_1 and a_2 between u and v in G . Define $\mathcal{M}_1 = (A, \mathcal{I}_1)$ the graphic matroid corresponding to G , and $\mathcal{M}_2 = (A, \mathcal{I}_2)$ the partition matroid in which independent sets are those with at most one arc incoming to every vertex $v \neq r$. In other words, we let

$$\mathcal{I}_2 = \{F : |F \cap \delta^-(v)| \leq 1 \text{ for all } v \in V \setminus \{r\}\},$$

where $\delta^-(v)$ denotes the set $\{(u, v) \in A\}$ of arcs incoming to v . Thus, any r -arborescence is independent in both matroids $\mathcal{M}_1$ and $\mathcal{M}_2$. Conversely, any set T independent in both $\mathcal{M}_1$ and $\mathcal{M}_2$ and of cardinality $|V| - 1$ (so that it is a base in both matroids) is an r -arborescence. Indeed, such a T being a spanning tree in G has a unique path between r and any vertex v ; this path must be directed from the root r since otherwise we would have either an arc incoming to r or two arcs incoming to the same vertex.

3 Algorithm for Maximum Cardinality Bipartite Matching

This section is largely taken from the lecture notes by Michel Goemans:

<http://math.mit.edu/~goemans/18433S09/matching-notes.pdf>

Recall the (unweighted) bipartite matching problem:

Definition 6 *Given a bipartite graph $G = (V, E)$, find a matching of maximum size.*

Recall that a matching $M \subseteq E$ is a subset of edges so that every vertex is incident to at most one edge of M , i.e., $|\{e \in M : v \in e\}| \leq 1$ for all $v \in V$.

Also recall that a path is a collection of edges $(v_0, v_1), (v_1, v_2), \dots, (v_{k-1}, v_k)$ where the v_i 's are distinct vertices. We can simply represent a path as $v_0 - v_1 - v_2 - \dots - v_k$.

Definition 7 (Alternating path) *An alternating path with respect to M is a path that alternates between edges in M and edges in $E \setminus M$.*

Definition 8 (Augmenting path) *An augmenting path with respect to M is an alternating path in which the first and last vertices are unmatched.*

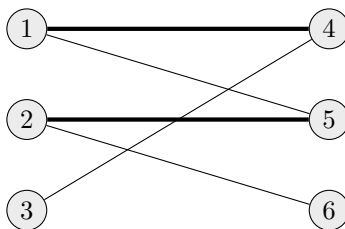


Figure 1: The edges $(3, 4), (4, 1), (1, 5), (5, 2), (2, 6)$ form an augmenting path

The definition of an augmenting path motivates the following algorithm:

AUGMENTINGPATHALGORITHM(G):
Input: A bipartite graph $G = (V, E)$.
Output: A matching M of maximum cardinality.

1. Initialize $M = \emptyset$.
2. **while** exists an augmenting path P
3. update $M = M \Delta P \equiv (M \setminus P) \cup (P \setminus M)$.
4. **return** M .

Exercise 1 *Devise an efficient algorithm for finding an augmenting path P (if one exists). What is the total running time of the AUGMENTINGPATHALGORITHM?*

In the next lecture we will prove that AUGMENTINGPATHALGORITHM indeed finds a maximum matching, and talk about duality (the vertex cover problem), which lets us certify optimality of the matching found.