Chapter 3

Matroid algorithms

The notion of matroids generalizes that of graphs and it is natural to ask which graph algorithms do have counterparts in the realm of matroids. The first issue to settle when talking about matroid algorithms is how an input matroid is given. The most common way of representing a matroid in an algorithm is using an oracle function that answers queries whether a given set of elements is independent. In this setting, we say that a matroid is given by an *oracle* and we assume that a single oracle query consumes a constant amount of time. As this assumption is often unrealistic (e.g., deciding independence for graphic matroids requires time linear in the size of the corresponding graph), we sometimes include in estimates on running times of algorithms the query time (see, e.g., Corollary 3.5). Then, it is easy to compute the actual running time of an algorithm by plugging in the query time to the estimate.

Let us also note that, in this lecture notes, we focus on establishing that problems of interest to us can be efficiently solved (i.e., solved in polynomial-time) without extensive optimization of the running time of the presented algorithms; the reader can find faster algorithms in the references we provide.

3.1 Minimum/maximum-weight base

The most classical algorithm related to matroids is the one for finding a base with the minimum or maximum-weight, the problem which includes the Minimum Spanning Tree (MST) problem for graphs. In the *maximum-weight base problem*, every element e of an input matroid $\mathcal{M}$ is assigned a weight $w(e)$ and the goal is to find a base B of $\mathcal{M}$ such that $\sum_{e\in B} w(e)$ is maximal. Such a base can be found by a greedy algorithm as shown in the next theorem.

Theorem 3.1. *The maximum-weight base problem can be solved by an algorithm running in time $O(n \log n + n\tau)$ for an n-element oracle-given matroid $\mathcal{M}$ with oracle query time τ.*

Proof. The problem can be solved by the following greedy algorithm: the elements of $\mathcal{M}$ are first sorted in a decreasing order based on their weight. Let $e_1, \ldots, e_n$ be the obtained order of the elements of $\mathcal{M}$.

The algorithm then constructs the base in n steps. At the beginning, set $A_0 = \emptyset$ and, in the i-th step, set A_i to $A_{i-1} \cup \{e_i\}$ if the set $A_{i-1} \cup \{e_i\}$ is independent, and to A_{i-1}, otherwise. Clearly, the set A_n is an inclusion-wise maximal independent set in $\mathcal{M}$ and thus A_n is a base of $\mathcal{M}$. The running time of the algorithm is bounded by $O(n \log n)$ needed for sorting the elements and $O(n\tau)$ for constructing the sets $A_1, \ldots, A_n$. Hence, the total running time of the algorithm is $O(n \log n + n\tau)$.

It remains to prove that the set A_n is a base of $\mathcal{M}$ with the maximum weight. Assume that there exists a base B of $\mathcal{M}$ with weight larger than the weight of A_n. Let r be the rank of $\mathcal{M}$, $i^a_1 < \cdots < i^a_r$ the indices of the elements contained in A_n and $i^b_1 < \cdots < i^b_r$ the indices of the elements of B. Since

$$\sum_{e \in A_n} w(e) < \sum_{e \in B} w(e) ,$$

there exists an index k such that $i^a_k < i^b_k$. Otherwise, it would hold that $w(e_{i^a_k}) \geq w(e_{i^b_k})$ for every $k = 1, \ldots, r$ and the sum of the weights of the elements of A_n would be at least as big as the sum of the weights of the elements of B.

Since the set $\{e_{i^a_1}, \ldots, e_{i^a_{k-1}}\}$ is an independent set of size $k-1$ in $\mathcal{M}$ and the set $\{e_{i^b_1}, \ldots, e_{i^b_k}\}$ is an independent set of size k, there exists an index j, $1 \leq j \leq k$, such that the set $\{e_{i^a_1}, \ldots, e_{i^a_{k-1}}, e_{i^b_j}\}$ is an independent set. However, the element $e_{i^b_j}$ should then have been included to the set $A_{i^b_j}$ (recall that $i^b_j \leq i^b_k < i^a_k$) which did not happen. This implies that there is no k such that $i^a_k < i^b_k$ and thus A_n is a maximum weight base of $\mathcal{M}$. □

The opposite problem, the *minimum-weight base* problem which asks for finding a base B with a minimal sum $\sum_{e \in B} w(e)$ can also be solved by a greedy algorithm. Note that the minimum-weight base problem includes the minimum spanning tree (MST) problem as a special case since spanning trees of a connected graph correspond to bases of the corresponding graphic matroid.

Corollary 3.2. *The minimum-weight base problem can be solved by an algorithm running in time $O(n \log n + n\tau)$ for an n-element oracle-given matroid $\mathcal{M}$ with oracle query time τ.*

Proof. Observe that if the weight of an element e of $\mathcal{M}$ is replaced with $-w(e)$, then a minimum-weight base with respect to the new weights is a maximum-weight base with respect to the original weights. The assertion of the corollary now follows from Theorem 3.1. □

Matroids are set systems for that the greedy algorithm finds a maximum-weight set in the system. However, this turns out to be yet another characterization of set systems that form independent sets of a matroid as stated in the next theorem.

Theorem 3.3. *Let $\mathcal{I}$ be a family of subsets of a set E that contains the empty set. If the greedy algorithm described in the proof of Theorem 3.1 finds a maximum-weight set in $\mathcal{I}$ for any non-negative weights $w(e)$, $e \in E$, then $(E, \mathcal{I})$ is a matroid. a matroid with ground set E.*

Proof. We verify that the family $\mathcal{I}$ satisfies the properties (I1), (I2) and (I3) given in the definition of a matroid. The empty set is contained in $\mathcal{I}$ by the assumption of the theorem which implies that (I1) holds.

We now show that (I2) also holds. Assume that (I2) fails and choose a set $I \in \mathcal{I}$ with the smallest number of elements such that not every subset of I is contained in $\mathcal{I}$. By the choice of I, there exists $I' \subseteq I$ with $|I \setminus I'| = 1$ such that $I' \notin \mathcal{I}$. Again, by the choice of I', every subset of I' is also contained in $\mathcal{I}$. Consider the following weight function on the elements of E:

$$w(e) = \begin{cases} 2 & \text{if } e \in I', \\ 1 & \text{if } e \in I \setminus I', \\ 0 & \text{otherwise.} \end{cases}$$

One of the sets in $\mathcal{I}$ with maximum weight is the set I with weight $2|I| - 1$. Hence, the greedy algorithm first sorts the elements, putting the elements of I' first, then the element of $I \setminus I'$ and the elements not contained in I as the last ones. If the greedy algorithm should output a set of weight of $2|I| - 1$, then it must include all elements of I' in the constructed maximum-weight set. Since $I' \notin \mathcal{I}$, the algorithm cannot include all these elements and thus it will not produce a set of weight $2|I| - 1$. We conclude that the set system $\mathcal{I}$ contains all subsets of every set in $\mathcal{I}$.

Assume now that the property (I3) does not hold and let I_1 and I_2 be the two sets violating it. Since $\mathcal{I}$ contains all subsets of every set, we can assume that $|I_2| = |I_1| + 1$. Consider the following weight function on the elements of E:

$$w(e) = \begin{cases} |I_1| + 2 & \text{if } e \in I_1, \\ |I_1| + 1 & \text{if } e \in I_2 \setminus I_1, \\ 0 & \text{otherwise.} \end{cases}$$

Since the sum of the weights of the elements of I_2 is at least $|I_2|(|I_1| + 1) \geq |I_1|^2 + 2|I_1| + 1$, the sum of the weights of the elements of a maximum-weight set is at least $|I_1|^2 + 2|I_1| + 1$. On the other hand, the greedy algorithm as described in the proof of Theorem 3.1 first constructs a set I_1 but it can then include no element of $I_2 \setminus I_1$ to I_1 as the set $I_1 + e$ is not independent for any $e \in I_2 \setminus I_1$. Hence, the weight of the set found by the greedy algorithm is $|I_1|(|I_1| + 2) = |I_1|^2 + 2|I_1|$.

This again contradicts the assumption that the greedy algorithm always finds a set with maximum weight. Since $\mathcal{I}$ satisfies all the three properties (I1), (I2) and (I3), $(E, \mathcal{I})$ forms a matroid as claimed in the statement in theorem. $\square$

3.2 Matroid intersection problem

Another graph algorithm that has a counterpart for matroids is the algorithm for constructing a maximum matching in bipartite graphs. Recall that a matching in a graph is a set of edges such that no two of them share a vertex. The problem of constructing a maximum-size matching in a bipartite graph can be formulated using matroids as follows: let G be a bipartite graph with no isolated vertices with parts V_1 and V_2 and edge set E. We define two matroids $\mathcal{M}_1$ and $\mathcal{M}_2$ with the ground set E. A subset $E' \subseteq E$ is independent in $\mathcal{M}_1$ if the end-vertices of the edges contained in E' are different in V_1. Analogously, E' is independent in $\mathcal{M}_2$ if the end-vertices of its edges in V_2 are different. The matroids $\mathcal{M}_1$ and $\mathcal{M}_2$ are transversal matroids. Observe that if G has no isolated vertices, then the rank of $\mathcal{M}_1$ is $|V_1|$ and the rank of $\mathcal{M}_2$ is $|V_2|$.

A subset E' of edges of G is a matching if and only if E' is independent in both $\mathcal{M}_1$ and $\mathcal{M}_2$. Hence, the problem of constructing a maximum-size matching can be formulated as the problem of constructing a maximum-size set independent in two given matroids with the same ground set. In the sequel, we show an efficient algorithm for solving this problem and find a minimax characterization of the size of such a maximum set. Let us start with the characterization (note that the sets E_1 and E_2 in the statement of the theorem can be assumed to be disjoint).

Theorem 3.4 (Matroid Intersection Theorem)**.** *If $\mathcal{M}_1$ and $\mathcal{M}_2$ are two matroids with the same ground set E and $\mathcal{I}_1$ and $\mathcal{I}_2$ are families of sets independent in $\mathcal{M}_1$ and $\mathcal{M}_2$, then the following equality holds:*

$$\max_{E' \in \mathcal{I}_1 \cap \mathcal{I}_2} |E'| = \min_{E_1 \cup E_2 = E} r_{\mathcal{M}_1}(E_1) + r_{\mathcal{M}_2}(E_2) \; .$$

Proof. We prove the equality as two inequalities. Let us start with the inequality "$\leq$". Consider two subsets E_1 and E_2 of E such that $E = E_1 \cup E_2$ and $r_{\mathcal{M}_1}(E_1) + r_{\mathcal{M}_2}(E_2)$ is minimized. We show that

$$|E'| \leq r_{\mathcal{M}_1}(E_1) + r_{\mathcal{M}_2}(E_2) \tag{3.1}$$

for every set $E' \subseteq E$ that is independent in both $\mathcal{M}_1$ and $\mathcal{M}_2$. As E' is independent in $\mathcal{M}_1$, we have that

$$|E' \cap E_1| \leq r_{\mathcal{M}_1}(E' \cap E_1) \leq r_{\mathcal{M}_1}(E_1) \tag{3.2}$$

Similarly, we derive that

$$|E' \cap E_2| \leq r_{\mathcal{M}_2}(E' \cap E_2) \leq r_{\mathcal{M}_2}(E_2) \tag{3.3}$$

Since $E = E_1 \cup E_2$, the estimates (3.2) and (3.3) yield (3.1) by the submodularity of the rank function. Since the inequality (3.1) holds for every $E' \subseteq E$, the inequality "$\leq$" of the statement of the theorem is established.

We now focus on establishing the opposite inequality. Fix $E' \subseteq E$ which is independent in both $\mathcal{M}_1$ and $\mathcal{M}_2$; such a set E' exists as the empty set is independent in both the matroids. We either establish the existence of a set E'' with $|E''| > |E'|$ that is independent in both $\mathcal{M}_1$ and $\mathcal{M}_2$ or find a partition of E into subsets E_1 and E_2 such that

$$|E'| = r_{\mathcal{M}_1}(E_1) + r_{\mathcal{M}_2}(E_2) \tag{3.4}$$

In order to achieve this goal, we first construct an auxiliary directed bipartite graph H.

One of the parts of H is formed by the elements of E' and the other part by the elements of $X = E \setminus E'$. The graph H contains an arc from $y \in E'$ to $x \in X$ if the set $(E' - y) + x$ is independent in $\mathcal{M}_1$ and an arc from $x \in X$ to $y \in E'$ if the set $(E' - y) + x$ is independent in $\mathcal{M}_2$. Set X_i to be the set of those $x \in X$ such that the set $E' + x$ is independent in $\mathcal{M}_i$, $i = 1, 2$. Our two cases are determined by the existence of a directed path from an element of X_1 to an element of X_2 in H.

Assume first that there is a directed path $x_0 y_1 x_1 \cdots y_\ell x_\ell$ in H from $x_0 \in X_1$ to $x_\ell \in X_2$. Consider such a shortest path. We allow ℓ to be equal to 0 (in which case $X_1 \cap X_2 \neq \emptyset$ and $E' + x_0$ is independent in both $\mathcal{M}_1$ and $\mathcal{M}_2$). We argue that the set $E'' = (E' \setminus \{y_1, \ldots, y_\ell\}) \cup \{x_0, \ldots, x_\ell\}$ is independent in both $\mathcal{M}_1$ and $\mathcal{M}_2$. By symmetry, it is enough to establish that E'' is independent in $\mathcal{M}_1$.

As the first step towards establishing the independence of E'' in $\mathcal{M}_1$, we prove that all the sets $E_k = (E' \setminus \{y_k, \ldots, y_\ell\}) \cup \{x_k, \ldots, x_\ell\}$ are independent in $\mathcal{M}_1$ for $k = 1, \ldots, \ell$. The proof proceeds by induction on k starting with $k = \ell$.

If $k = \ell$, then the set $E_k = E_\ell = (E' - y_\ell) + x_\ell$ is independent in $\mathcal{M}_1$ since H contains an arc from y_ℓ to x_ℓ. Assume now that we have already established that the sets $E_{k+1}, \ldots, E_\ell$ are independent in $\mathcal{M}_1$ and we want to show that the set E_k is also independent in $\mathcal{M}_1$. As $E_k - x_k \subseteq E_{k+1}$, the set $E_k - x_k$ is independent in $\mathcal{M}_1$ and its size is $|E'| - 1$. Since H contains an arc from y_k to x_k, the set $(E' - y_k) + x_k$ is an independent set in $\mathcal{M}_1$ of size $|E'|$. Hence, there exists $z \in (E' - y_k) + x_k$ such that $z \notin E_k - x_k$ and the set $(E_k - x_k) + z$ is independent in $\mathcal{M}_1$. If $z = x_k$, then E_k is independent in $\mathcal{M}_1$ and the proof is finished. Suppose that $z \neq x_k$, i.e., $z = y_i$ for $i = k+1, \ldots, \ell$ as

$$((E' - y_k) + x_k) \setminus (E_k - x_k) = \{x_k, y_{k+1}, \ldots, y_\ell\} \, .$$

Since $(E_k - x_k) + y_i$ is an independent set of size $|E'|$ in $\mathcal{M}_1$, there exists $z' \in (E_k - x_k) + y_i$ such that $z' \notin E' - y_k$ and $(E' - y_k) + z'$ is an independent set in $\mathcal{M}_1$. Observe that z' must be one of the elements $x_{k+1}, \ldots, x_\ell$ as all the other elements of $(E_k - x_k) + y_i$ are contained in $E' - y_k$. In particular, H contains an arc from y_k to $z' = x_{i'}$, $i' > k$, contradicting the choice of the path from x_0 to x_ℓ as the shortest possible.

We have now derived that the set $(E' \setminus \{y_1, \dots, y_\ell\}) \cup \{x_1, \dots, x_\ell\}$ is independent in $\mathcal{M}_1$. Since the set $E' + x_i$ for $i = 1, \dots, \ell$ is not independent in $\mathcal{M}_1$ by the choice of the path from x_0 to x_ℓ as the shortest path from X_1 to X_2, the rank of the set $E' \cup \{x_1, \dots, x_\ell\}$ in $\mathcal{M}_1$ is equal to $|E'|$. Finally, the rank of the set $E' \cup \{x_0, x_1, \dots, x_\ell\}$ is equal to $|E'| + 1$ because the set $E' \cup \{x_0\}$ is independent in $\mathcal{M}_1$. Hence, the rank of the set $E'' = (E' \setminus \{y_1, \dots, y_\ell\}) \cup \{x_0, x_1, \dots, x_\ell\}$ must be $|E'| + 1$, i.e., the set E'' is independent in $\mathcal{M}_1$ as desired. As mentioned earlier, a symmetric argument yields that E'' is independent in $\mathcal{M}_2$, too. This finishes the case when there is a directed path from an element of X_1 to an element of X_2 in the auxiliary graph H.

Assume now that there is no directed path from an element of X_1 to an element of X_2. We aim to partition E into two subsets E_1 and E_2 such that (3.4) holds. As there is no directed path from X_1 to X_2, the vertices of H (and thus the corresponding elements of E) can be partitioned into two sets E_1 and E_2 such that $X_1 \subseteq E_2$, $X_2 \subseteq E_1$ and there is no arc leading from E_2 to E_1. We claim that

$$r_{\mathcal{M}_1}(E_1) = |E' \cap E_1| \tag{3.5}$$

and

$$r_{\mathcal{M}_2}(E_2) = |E' \cap E_2| \tag{3.6}$$

We focus on proving the equation (3.5). The equation (3.6) follows the lines of the proof of (3.5) through replacing the roles of $\mathcal{M}_1$ and $\mathcal{M}_2$ and reversing the orientation of the arcs of H.

Assume that $r_{\mathcal{M}_1}(E_1) > |E' \cap E_1|$. Hence, there exists $x \in E_1 \setminus E'$ such that the set $(E' \cap E_1) + x$ is independent in $\mathcal{M}_1$. Clearly, the set $(E' \cap E_1) + x$ can be completed by elements of E' to a set of size $|E'|$ independent in $\mathcal{M}_1$. Observe that the resulting set must be equal to $(E' - y) + x$ for some $y \in E' \cap E_2$. Hence, H contains an arc from y to x that contradicts our choice of E_1 and E_2 as two sets such that there is no arc from E_2 to E_1. This establishes (3.5). Since the equality (3.6) follows from a symmetric argument, the desired equality (3.4) is now established. □

As a corollary of Theorem 3.4, we obtain that the problem of determining and constructing the maximum-size set independent in two given matroids, which is known as the *matroid intersection problem*, can be solved in polynomial-time.

Corollary 3.5. *The matroid intersection problem is solvable in time $O(r^2 n\tau)$ for two oracle-given matroids of rank at most r with at most n elements and with oracle query time τ.*

Proof. The proof of Theorem 3.4 suggests an algorithm for constructing a maximum-size set independent in both the input matroids starting with the empty set and augmenting it in each phase by one element. The number of phases of such an algorithm is bounded by r. In each phase, the algorithm queries $2|E'| \cdot |E \setminus E'| \leq 2rn = O(rn)$ times the oracles to construct the auxiliary graph H and $2|E \setminus E'| \leq 2n$ times to construct the sets X_1 and X_2. The algorithm then checks the existence of a directed path from X_1 to X_2 in time $O(rn)$ and if it exists it augments the constructed independent set. Otherwise, it stops. It is straightforward to estimate the running time of this algorithm by $O(r^2 n\tau)$. $\square$

We remark here that there are faster algorithms for solving the matroid intersection problem, e.g., there is one running in time $O(r^{3/2} n\tau)$ [4].