2.1 Minimum Spanning Trees

Let $G = (V, E)$ be a connected undirected graph.

Definition 2.4 A *forest* in G is a subgraph $F = (V, E')$ with no cycles. Note that F has the same vertex set as G. A *spanning tree* in G is a forest with exactly one connected component. Given weights $w : E \to \mathcal{N}$ (edges are assigned weights over the natural numbers), a *minimum (weight) spanning tree (MST)* in G is a spanning tree T whose total weight (sum of the weights of the edges in T) is minimum over all spanning trees. □

Lemma 2.5 *Let $F = (V, E)$ be an undirected graph, c the number of connected components of F, $m = |E|$, and $n = |V|$. Then F has no cycles iff $c + m = n$.*

Proof.
($\rightarrow$) By induction on m. If $m = 0$, then there are n vertices and each forms a connected component, so $c = n$. If an edge is added without forming a cycle, then it must join two components. Thus m is increased by 1 and c is decreased by 1, so the equation $c + m = n$ is maintained.

($\leftarrow$) Suppose that F has at least one cycle. Pick an arbitrary cycle and remove an edge from that cycle. Then m decreases by 1, but c and n remain the same. Repeat until there are no more cycles. When done, the equation $c + m = n$ holds, by the preceding paragraph; but then it could not have held originally. □

We use a *greedy algorithm* to produce a minimum weight spanning tree. This algorithm is originally due to Kruskal [66].

Algorithm 2.6 (Greedy Algorithm for MST)

1. Sort the edges by weight.
2. For each edge on the list in order of increasing weight, include that edge in the spanning tree if it does not form a cycle with the edges already taken; otherwise discard it.

The algorithm can be halted as soon as $n - 1$ edges have been kept, since we know we have a spanning tree by Lemma 2.5.

Step 1 takes time $O(m \log m) = O(m \log n)$ using any one of a number of general sorting methods, but can be done faster in certain cases, for example if the weights are small integers so that bucket sort can be used.

Later on, we will give an almost linear time implementation of step 2, but for now we will settle for $O(n \log n)$. We will think of including an edge e in the spanning tree as taking the union of two disjoint sets of vertices, namely the vertices in the connected components of the two endpoints of e in the forest being built. We represent each connected component as a linked list. Each list element points to the next element and has a back pointer to the head of the list. Initially there are no edges, so we have n lists, each containing one vertex. When a new edge (u, v) is encountered, we check whether it would form a cycle, *i.e.* whether u and v are in the same connected component, by comparing back pointers to see if u and v are on the same list. If not, we add (u, v) to the spanning tree and take the union of the two connected components by merging the two lists. Note that the lists are always disjoint, so we don't have to check for duplicates.

Checking whether u and v are in the same connected component takes constant time. Each merge of two lists could take as much as linear time, since we have to traverse one list and change the back pointers, and there are $n - 1$ merges; this will give $O(n^2)$ if we are not careful. However, if we maintain counters containing the size of each component and always merge the smaller into the larger, then each vertex can have its back pointer changed at most $\log n$ times, since each time the size of its component at least doubles. If we charge the change of a back pointer to the vertex itself, then there are at most $\log n$ changes per vertex, or at most $n \log n$ in all. Thus the total time for all list merges is $O(n \log n)$.

2.2 The Blue and Red Rules

Here is a more general approach encompassing most of the known algorithms for the MST problem. For details and references, see [100, Chapter 6], which proves the correctness of the greedy algorithm as a special case of this more general approach. In the next lecture, we will give an even more general treatment.

Let $G = (V, E)$ be an undirected connected graph with edge weights $w : E \to \mathcal{N}$. Consider the following two rules for coloring the edges of G, which Tarjan [100] calls the *blue rule* and the *red rule*:

Blue Rule: Find a *cut* (a partition of V into two disjoint sets X and $V - X$) such that no blue edge crosses the cut. Pick an uncolored edge of minimum weight between X and $V - X$ and color it blue.
Red Rule: Find a *cycle* (a path in G starting and ending at the same vertex) containing no red edge. Pick an uncolored edge of maximum weight on that cycle and color it red.

The greedy algorithm is just a repeated application of a special case of the blue rule. We will show next time:

Theorem 2.7 *Starting with all edges uncolored, if the blue and red rules are applied in arbitrary order until neither applies, then the final set of blue edges forms a minimum spanning tree.*

Lecture 3 Matroids and Independence

Before we prove the correctness of the blue and red rules for MST, let's first discuss an abstract combinatorial structure called a *matroid.* We will show that the MST problem is a special case of the more general problem of finding a minimum-weight maximal independent set in a matroid. We will then generalize the blue and red rules to arbitrary matroids and prove their correctness in this more general setting. We will show that every matroid has a dual matroid, and that the blue and red rules of a matroid are the red and blue rules, respectively, of its dual. Thus, once we establish the correctness of the blue rule, we get the red rule for free.

We will also show that a structure is a matroid if and only if the greedy algorithm always produces a minimum-weight maximal independent set for any weighting.

Definition 3.1 A *matroid* is a pair $(S, \mathcal{I})$ where S is a finite set and $\mathcal{I}$ is a family of subsets of S such that

(i) if $J \in \mathcal{I}$ and $I \subseteq J$, then $I \in \mathcal{I}$;

(ii) if $I, J \in \mathcal{I}$ and $|I| < |J|$, then there exists an $x \in J - I$ such that $I \cup \{x\} \in \mathcal{I}$.

The elements of $\mathcal{I}$ are called *independent sets* and the subsets of S not in $\mathcal{I}$ are called *dependent sets.* □

This definition is supposed to capture the notion of *independence* in a general way. Here are some examples:

1. Let V be a vector space, let S be a finite subset of V, and let $\mathcal{I} \subseteq 2^S$ be the family of linearly independent subsets of S. This example justifies the term "independent".

2. Let A be a matrix over a field, let S be the set of rows of A, and let $\mathcal{I} \subseteq 2^S$ be the family of linearly independent subsets of S.

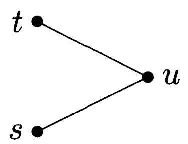
3. Let $G = (V, E)$ be a connected undirected graph. Let $S = E$ and let $\mathcal{I}$ be the set of forests in G. This example gives the MST problem of the previous lecture.

4. Let $G = (V, E)$ be a connected undirected graph. Let $S = E$ and let $\mathcal{I}$ be the set of subsets $E' \subseteq E$ such that the graph $(V, E - E')$ is connected.

5. Elements $\alpha_1, \ldots, \alpha_n$ of a field are said to be *algebraically independent* over a subfield k if there is no nontrivial polynomial $p(x_1, \ldots, x_n)$ with coefficients in k such that $p(\alpha_1, \ldots, \alpha_n) = 0$. Let S be a finite set of elements and let $\mathcal{I}$ be the set of subsets of S that are algebraically independent over k.

Definition 3.2 A *cycle* (or *circuit*) of a matroid $(S, \mathcal{I})$ is a setwise minimal (*i.e.*, minimal with respect to set inclusion) dependent set. A *cut* (or *cocircuit*) of $(S, \mathcal{I})$ is a setwise minimal subset of S intersecting all maximal independent sets. □

The terms *circuit* and *cocircuit* are standard in matroid theory, but we will continue to use *cycle* and *cut* to maintain the intuitive connection with the special case of MST. However, be advised that cuts in graphs as defined in the last lecture are *unions* of cuts as defined here. For example, in the graph



the set $\{(s, u),\ (t, u)\}$ forms a cut in the sense of MST, but not a cut in the sense of the matroid, because it is not minimal. However, a moment's thought reveals that this difference is inconsequential as far as the blue rule is concerned.

Let the elements of S be weighted. We wish to find a setwise maximal independent set whose total weight is minimum among all setwise maximal independent sets. In this more general setting, the blue and red rules become:

Blue Rule: Find a cut with no blue element. Pick an uncolored element of the cut of minimum weight and color it blue. **Red Rule:** Find a cycle with no red element. Pick an element of the cycle of maximum weight and color it red.

3.1 Matroid Duality

As the astute reader has probably noticed by now, there is some kind of duality afoot. The similarity between the blue and red rules is just too striking to be mere coincidence.

Definition 3.3 Let $(S,\mathcal{I})$ be a matroid. The *dual matroid* of $(S,\mathcal{I})$ is $(S,\mathcal{I}^*)$, where

$$\mathcal{I}^* \quad = \quad \{\text{subsets of } S \text{ disjoint from some maximal element of } \mathcal{I}\} \ .$$

In other words, the maximal elements of $\mathcal{I}^*$ are the complements in S of the maximal elements of $\mathcal{I}$. □

The examples 3 and 4 above are duals. Note that $\mathcal{I}^{**} = \mathcal{I}$. Be careful: it is *not* the case that a set is independent in a matroid iff it is dependent in its dual. For example, except in trivial cases, $\emptyset$ is independent in both matroids.

Theorem 3.4

1. *Cuts in $(S,\mathcal{I})$ are cycles in $(S,\mathcal{I}^*)$.*
2. *The blue rule in $(S,\mathcal{I})$ is the red rule in $(S,\mathcal{I}^*)$ with the ordering of the weights reversed.*

3.2 Correctness of the Blue and Red Rules

Now we prove the correctness of the blue and red rules in arbitrary matroids. A proof for the special case of MST can be found in Tarjan's book [100, Chapter 6]; Lawler [70] states the blue and red rules for arbitrary matroids but omits a proof of correctness.

Definition 3.5 Let $(S,\mathcal{I})$ be a matroid with dual $(S,\mathcal{I}^*)$. An *acceptable coloring* is a pair of disjoint sets $B \in \mathcal{I}$ (the *blue elements*) and $R \in \mathcal{I}^*$ (the *red elements*). An acceptable coloring B, R is *total* if $B \cup R = S$, *i.e.* if B is a maximal independent set and R is a maximal independent set in the dual. An acceptable coloring B', R' *extends* or *is an extension of* an acceptable coloring B, R if $B \subseteq B'$ and $R \subseteq R'$. □

Lemma 3.6 *Any acceptable coloring has a total acceptable extension.*

Proof. Let B, R be an acceptable coloring. Let U^* be a maximal element of $\mathcal{I}^*$ extending R, and let $U = S - U^*$. Then U is a maximal element of $\mathcal{I}$ disjoint from R. As long as $|B| < |U|$, select elements of U and add them to B, maintaining independence. This is possible by axiom (ii) of matroids. Let $\widehat{B}$ be the resulting set. Since all maximal independent sets have the same cardinality (Exercise 1a, Homework 1), $\widehat{B}$ is a maximal element of $\mathcal{I}$ containing B and disjoint from R. The desired total extension is $\widehat{B}, S - \widehat{B}$. □

Lemma 3.7 *A cut and a cycle cannot intersect in exactly one element.*

Proof. Let C be a cut and D a cycle. Suppose that $C \cap D = \{x\}$. Then $D - \{x\}$ is independent and $C - \{x\}$ is independent in the dual. Color $D - \{x\}$ blue and $C - \{x\}$ red; by Lemma 3.6, this coloring extends to a total acceptable coloring. But depending on the color of x, either C is all red or D is all blue; this is impossible in an acceptable coloring, since D is dependent and C is dependent in the dual. $\square$

Suppose B is independent and $B \cup \{x\}$ is dependent. Then $B \cup \{x\}$ contains a minimal dependent subset or cycle C, called the *fundamental cycle*[1] of x and B. The cycle C must contain x, because $C - \{x\}$ is contained in B and is therefore independent.

Lemma 3.8 (Exchange Lemma) *Let B, R be a total acceptable coloring.*

(i) Let $x \in R$ and let y lie on the fundamental cycle of x and B. If the colors of x and y are exchanged, the resulting coloring is acceptable.

(ii) Let $y \in B$ and let x lie on the fundamental cut of y and R (the fundamental cut of y and R is the fundamental cycle of y and R in the dual matroid). If the colors of x and y are exchanged, the resulting coloring is acceptable.

Proof. By duality, we need only prove (i). Let C be the fundamental cycle of x and B and let y lie on C. If $y = x$, there is nothing to prove. Otherwise $y \in B$. The set $C - \{y\}$ is independent since C is minimal. Extend $C - \{y\}$ by adding elements of $|B|$ as in the proof of Lemma 3.6 until achieving a maximal independent set B'. Then $B' = (B - \{y\}) \cup \{x\}$, and the total acceptable coloring $B', S - B'$ is obtained from B, R by switching the colors of x and y. $\square$

A total acceptable coloring B, R is called *optimal* if B is of minimum weight among all maximal independent sets; equivalently, if R is of maximum weight among all maximal independent sets in the dual matroid.

Lemma 3.9 *If an acceptable coloring has an optimal total extension before execution of the blue or red rule, then so has the resulting coloring afterwards.*

Proof. We prove the case of the blue rule; the red rule follows by duality. Let B, R be an acceptable coloring with optimal total extension $\widehat{B}, \widehat{R}$. Let A be a cut containing no blue elements, and let x be an uncolored element of A of minimum weight. If $x \in \widehat{B}$, we are done, so assume that $x \in \widehat{R}$. Let C be the fundamental cycle of x and $\widehat{B}$. By Lemma 3.7, $A \cap C$ must contain

[1] We say "the" because it is unique (Exercise 1b, Homework 1), although we do not need to know this for our argument.

another element besides x, say y. Then $y \in \widehat{B}$, and $y \notin B$ because there are no blue elements of A. By Lemma 3.8, the colors of x and y in $\widehat{B}, \widehat{R}$ can be exchanged to obtain a total acceptable coloring $\widehat{B}', \widehat{R}'$ extending $B \cup \{x\}, R$. Moreover, $\widehat{B}'$ is of minimum weight, because the weight of x is no more than that of y. □

We also need to know

Lemma 3.10 *If an acceptable coloring is not total, then either the blue or red rule applies.*

Proof. Let B, R be an acceptable coloring with uncolored element x. By Lemma 3.6, B, R has a total extension $\widehat{B}, \widehat{R}$. By duality, assume without loss of generality that $x \in \widehat{B}$. Let C be the fundamental cut of x and $\widehat{R}$. Since all elements of C besides x are in $\widehat{R}$, none of them are blue in B. Thus the blue rule applies. □

Combining Lemmas 3.9 and 3.10, we have

Theorem 3.11 *If we start with an uncolored weighted matroid and apply the blue or red rules in any order until neither applies, then the resulting coloring is an optimal total acceptable coloring.*

What is really going on here is that all the subsets of the maximal independent sets of minimal weight form a submatroid of $(S, \mathcal{I})$, and the blue rule gives a method for implementing axiom (ii) for this matroid; see Miscellaneous Exercise 1.

3.3 Matroids and the Greedy Algorithm

We have shown that if $(S, \mathcal{I})$ is a matroid, then the greedy algorithm produces a maximal independent set of minimum weight. Here we show the converse: if $(S, \mathcal{I})$ is not a matroid, then the greedy algorithm fails for some choice of integer weights. Thus the abstract concept of matroid captures exactly when the greedy algorithm works.

Theorem 3.12 ([32]; see also [70]) *A system $(S, \mathcal{I})$ satisfying axiom (i) of matroids is a matroid (i.e., it satisfies (ii)) if and only if for all weight assignments $w : S \rightarrow \mathcal{N}$, the greedy algorithm gives a minimum-weight maximal independent set.*

Proof. The direction $(\rightarrow)$ has already been shown. For $(\leftarrow)$, let $(S, \mathcal{I})$ satisfy (i) but not (ii). There must be A, B such that $A, B \in \mathcal{I}$, $|A| < |B|$, but for no $x \in B - A$ is $A \cup \{x\} \in \mathcal{I}$.

Assume without loss of generality that B is a *maximal* independent set. If it is not, we can add elements to B maintaining the independence of B; for any element that we add to B that can also be added to A while preserving the independence of A, we do so. This process never changes the fact that $|A| < |B|$ and for no $x \in B - A$ is $A \cup \{x\} \in \mathcal{I}$.

Now we assign weights $w : S \to \mathcal{N}$. Let $a = |A - B|$ and $b = |B - A|$. Then $a < b$. Let h be a huge number, $h \gg a, b$. (Actually $h > b^2$ will do.)

Case 1 If A is a maximal independent set, assign

$$\begin{array}{ll} w(x) = a + 1 & \text{for } x \in B - A \\ w(x) = b + 1 & \text{for } x \in A - B \\ w(x) = 0 & \text{for } x \in A \cap B \\ w(x) = h & \text{for } x \notin A \cup B \ . \end{array}$$

Thus

$$\begin{aligned} w(A) &= a(b+1) &= ab + a \\ w(B) &= b(a+1) &= ab + b \ . \end{aligned}$$

This weight assignment forces the greedy algorithm to choose B when in fact A is a maximal independent set of smaller weight.

Case 2 If A is not a maximal independent set, assign

$$\begin{array}{ll} w(x) = 0 & \text{for } x \in A \\ w(x) = b & \text{for } x \in B - A \\ w(x) = h & \text{for } x \notin A \cup B \ . \end{array}$$

All the elements of A will be chosen first, and then a huge element outside of $A \cup B$ must be chosen, since A is not maximal. Thus the minimum-weight maximal independent set B was not chosen. $\square$