Matroids

IE 496 Lecture 16

Reading for This Lecture

- Kozen, Lecture 3
- Paper on Beruka's Algorithm

Prim's Algorithm

```
S is the set of nodes in the tree

S = {0}
for (i = 0; i < n; i++){

  SELECT i ∉ S nearest to S;
  S = UNION(S, i);
}
```

Kruskal's Algorithm

T is the set of edges in the tree

```
T = Ø
for (i = 0; i < m; i++){
  SELECT the cheapest edge e
  if (feasible(UNION(T, e)){
     UNION(T, e);
}
```

The Red and Blue Rules

- Start with all edges uncolored
- The Blue Rule:
  - Find a cut with no BLUE edges.
  - Pick an edge of minimum weight in the cut and color it BLUE.
- The Red Rule:
  - Find a cycle containing no RED edges.
  - Pick an uncolored edge of maximum weight and color it RED.
- Arbitrary application of the Red and Blue rules will result in a minimum spanning tree (blue edges).

Matroids

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

  (i) If $J \in \mathscr{I}$ and $I \subseteq J$, then $I \in \mathscr{I}$

  (ii) If I, J and $|I| < |J|$, then there exists an $x \in J \setminus I$ such that $I \cup \{x\} \in \mathscr{I}$

- Elements of $\mathscr{I}$ are called the *independent sets*.
- Note that all independent sets have the same cardinality.
- A *cycle* is a setwise minimal dependent set.
- A *cut* is a setwise minimal subset of S intersecting all maximal independent sets.

Matroid Examples

- Graph $G = (V, E)$
  - $\mathscr{I}$ is the set of forests in G (acyclic subgraphs).
- Vector space V
  - $\mathscr{I}$ is the set of all linearly independent subsets of V.
- Columns/rows of a matrix A
  - $\mathscr{I}$ is the set of all bases of A.

Importance of Matroids

- Why study matroids?
- Matroids are common mathematical structures.
- In a matroid, we can always find the minimum-weight maximal independent set using the greedy algorithm.
- Algorithm: Apply the Red and Blue rules arbitrarily.
- In fact, $(S, \mathscr{I})$ satisfying property (i) is a matroid if and only if we can find a minimum-weight maximal independent set using the greedy algorithm!

Matroid duality

- The dual of a matroid $(S, \mathscr{I})$ is $(S, \mathscr{I}^*)$ where

  $\mathscr{I}^* = \{S' \subseteq S$ disjoint from some maximal element of $\mathscr{I}\}$
- The maximal elements of $\mathscr{I}^*$ are the complements of the maximal elements of $\mathscr{I}$.
- Properties
  - Cuts in $(S, \mathscr{I})$ are cycles in $(S, \mathscr{I}^*)$.
  - The blue rule in $(S, \mathscr{I})$ is the red rule in $(S, \mathscr{I}^*)$ with the weights reversed.

Boruvka's Algorithm

- At each step, select all edges that connect some component of the graph to it's nearest neighbor.
- Add all these edges to the tree simultaneously.
- Why does this work?

- Sequential Implementation

Component Labeling

- Given a graph G = (V, E).
- Component labeling is numbering each vertex according to which component it belongs to.
- Sequential Component Labeling Algorithms
  - Breadth-first search
  - Union-find
- This is the same as finding the equivalence classes in an ordered set.

Parallel Component Labeling

- Algorithm

- Analysis

Parallelizing Boruvka's Algorithm

Algorithm

- Initialization
  - Find minimum edge adjacent to each node and mark them.
- Iterate
  - Perform parallel component labeling.
  - Find minimum edge connecting each node to another component.
  - Find overall minimum edge connecting each component to another.
  - Add all these edges into the graph