

Day 8 Notes

Zach Neveu

May 16, 2019

1 Agenda

- Matching concepts
- Algorithm to solve matching
- Bipartite matching
- Intro to linear programming

2 Matching Review

- Review of matching problem from last class
- Swapping membership of augmenting path yields larger matching
- Maximum matching includes all nodes
- Key result: given a matching, M , the matching is optimal if and only if no augmenting path with respect to M exists.
- If no augmenting path $\rightarrow M$ is optimal (not so obvious)
- If M optimal \rightarrow no augmenting path (this is fairly straightforward)

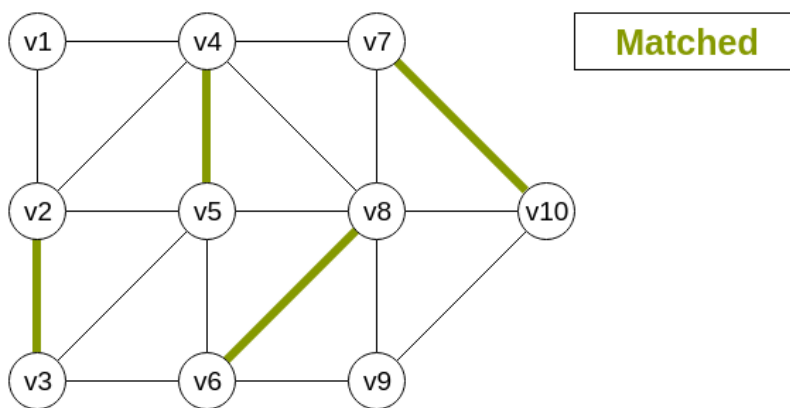


Figure 1: Review matching diagram from Day 6

Matching Algorithm

```
def match(g):  
    m = 0  
    while p = aug_path(M):  
        swap_membership(p)
```

- Outer loop runs $O(E)$ times where E is edges
- Time to find an augmenting path is polynomial time
- Algorithm to find augmenting path is convoluted and not particularly useful to know

Bipartite Matching

- **Bipartite Graph:** a graph where the nodes can be divided into two groups such that every edge goes from one group to the other (no edges are inside a group).
- Bipartite Matching: find the largest matching in a bipartite graph
- Classic example problem: Job scheduling on heterogeneous computers
 - Group of jobs that all need to be done
 - Group of computers that can run jobs
 - Certain jobs can only run on certain computers
 - Find how to get the most jobs done

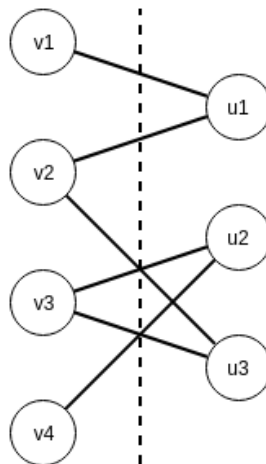


Figure 2: Bipartite Graph Example: Every edge crosses dividing line

- Start at all exposed nodes
- Do BFS across alternating edges until another exposed node is reached
- The path between exposed nodes that is found is an augmenting path
- BFS runs in fast polynomial time and finds any augmenting paths that exist

- Search diagram has structure: matched stages don't branch, matched and unmatched stages alternate
- Why is this like greedy algorithms?
 - Matching always getting bigger
 - Not quite greedy: some edges get deleted when swapping membership
 - Kind of a “deeper” greedy algorithm
 - Approach applicable to many problems

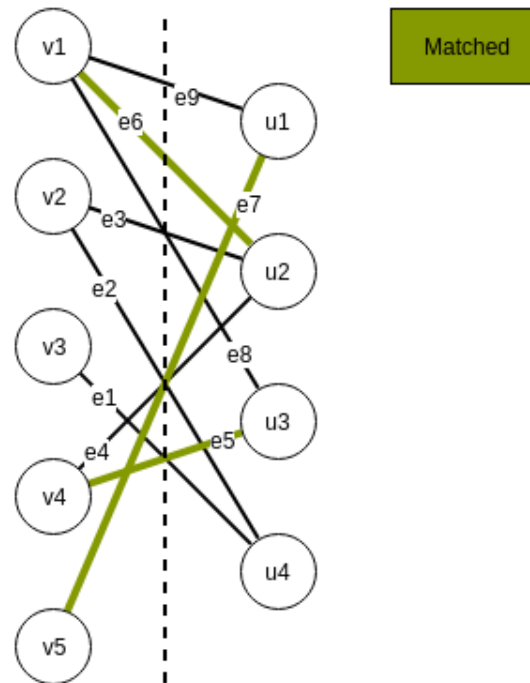


Figure 3: Exaple Bipartite matching

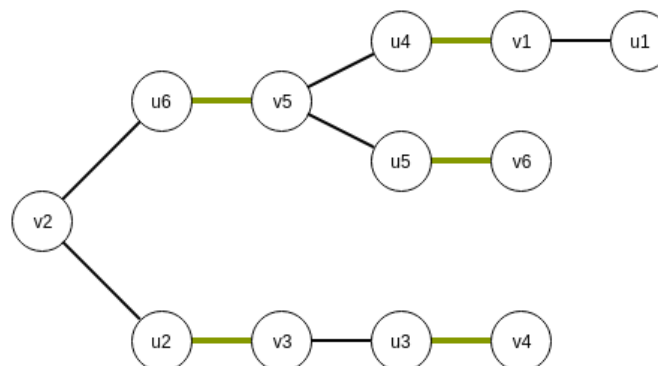


Figure 4: Example search diagram (different problem from fig. 3)

3 Linear Integer Programming (LP)

- Many problems have special format
- If your problem can be re-phrased into this format, it can be solved FAST by existing solvers.
- Leverage genius that is not yours
- Not super well known in ECE, came from applied math
- Large LP instances solvable in minutes
- Trade-off is that problem must be in exact format

Example

- You are a politician trying to win an election. Your district has 3 regions:
 - Urban - 100k voters
 - Suburban - 200k voters
 - Rural - 50k voters
- Goal is to get a majority in each region
- Win votes by advertising based on 4 issues
 - Building roads
 - Gun control
 - Farm subsidies
 - Gas taxes
- Goal: get max results within advertising budget

Table 1: Problem Breakdown Table

	urban	suburban	rural
Build roads	0.2	5	3
gun control	8	2	-5
farm subsidies	0	0	10
gas taxes	10	0	-2