CS 561, HW7

Prof. Jared Saia, University of New Mexico

Due: Dec 5th

1. Exercise 26.2-4: Minimum cut corresponding to maximum flow

2. Exercise 26.2-11 : Edge Connectivity

3. A bipartite graph is a graph that contains no cycle with an odd number of edges. Recall from the wrestler problem that a graph, $G = (V, E)$ is bipartite iff V can be partitioned into two sets L, R such that all edges in E have one endpoint in L and one endpoint in R. A *matching* in a graph is a set of edges $E' \subseteq E$ such that each vertex in V is matched at most once, i.e. it is incident to a most one edge in E'. A *perfect matching* is a matching where every vertex is matched, i.e. is incident to exactly one edge in E'. For a set of vertices $S \subseteq V$, let $N(S)$ be the set of all neighbors of S, i.e. $\{y \in V : (x, y) \in E \text{ for some } x \in S\}$

   Assume we are given a bipartite graph where $|L| = |R|$. Prove that there is a perfect matching in G iff $|N(S)| \geq |S|$ for all $S \subseteq L$.

   Hint: Use the Max flow/Min Cut Theorem

4. Consider the following problem related to segmenting the pixels of an image between foreground and background. We have a picture that consists of n pixels. We represent this as an undirected graph $G = (V, E)$ where V is the set of pixels and there is an edge $(i, j) \in E$ iff pixel i and pixel j are neighbors in the image.[1] We want to find a good segmentation, which is an assignment of each pixel to either the foreground or the background.

   For each pixel i, we have a likelihood a_i that i belongs to the foreground and a likelihood b_i that i belongs to the background. These likelihood values are all non-negative. Additionally, for each edge $(i, j) \in E$, we

[1] Note that we'd commonly expect this graph to be a grid, but in fact we want to handle any arbitrary graph (to handle, e.g., 3-D images, wrapped are warped images, etc)

have a non-negative separation penalty $p_{i,j}$ which is charged if one of i or j is assigned to the foreground and the other is assigned to the background.

Our problem then is to find a partition of the set of pixels into sets A and B so as to maximize:

$$L(A, B) = \sum_{i \in A} a_i + \sum_{i \in B} b_i - \sum_{(i,j) \in E, |A \cap \{i,j\}|=1} p_{i,j}$$

.

Give an efficient algorithm to solve this problem.

5. Exercise 29.2-2 (Linear Program for the example in Figure 24.2(a))

6. Exercise 29.2-4 (Network Flow as an LP)

7. Rock, Paper, Scissors is a simple 2 person game. In a given round, both players simultaneously choose either Rock, Paper or Scissors. If they both choose the same object, it's a tie. Otherwise, Rock beats Scissors; Scissors beats Paper; and Paper beats Rock. Imagine you're playing the following betting variant of this game with a friend. When Scissors beats Paper, or Paper beats Rock, the loser gives the winner \$1. However, in the case when Rock beats Scissors, this is called a SMASH, and the loser must give the winner \$10.

   (a) Say you know that your friend will choose Rock, Scissors or Paper, each with probability 1/3. Write a linear program to calculate the probabilities you should use of choosing each object in order to maximize your expected winnings. Let p_1, p_2, p_3 be variables associated with the best way of choosing Rock, Scissors and Paper respectively. Note: If you want to check your work, there are several free linear program solvers on the Internets: check the Wikipedia page on linear programming.

   (b) Now say that your friend is smart and, also, clairvoyant: she will magically know the exact probabilities you are using and will respond optimally. Write another linear program to calculate the probabilities you should now use in order to maximize your expected winnings. Hint 1: If your opponent knows your strategy, her strategy will be to choose one of the three objects with probability 1. Hint 2: Review the LP you wrote for the shortest paths problem.

8. The problem INDEPENDENT-SET asks: Does there exist a set of k vertices in a graph G with no edges between them?. Show that this problem is NP-Complete. (hint: Reduce from CLIQUE)