CSC 2420 Fall 2012, Assignment 2
Due: November 15 at start of class

1. As briefly discussed in class, Arkin and Silverberg [1987] reduce the m machine weighted interval scheduling problem to a min cost flow problem. Namely they transform an instance $\mathcal{I}$ of the interval scheduling problem into an instance of a flow problem $\mathcal{F}_{\mathcal{I}}$ and argue that removing a least valuable set of intervals from $\mathcal{I}$ so that all remaining intervals can be scheduled is equiavlent to finding a least cost flow for flow value $f = \chi(\mathcal{I}) - m$ where $\chi(\mathcal{I})$ is the size of a maximum clique in the interval graph induced by $\chi(\mathcal{I})$.

   Provide a convincing argument to fill in the details of this result.

   You can download the Arkin and Silverberg papere or ask me if you have trouble locating it. (Note: I am not using their notation and in particular I use m where they use k.)

2. Consider the maximum matching problem. That is, given a graph $G = (V, E)$, find a subset of edges $E' \subseteq E$ such that for all nodes $u \in V$, the degree of u in $G' = (V, E')$ is at most 1. Let $IN(u) = \{e : e = (u, v) \in E \text{ for some } v \in V\}$. We can express the maximum matching problem as the following natural IP:
   maximize $\sum_{e \in E} x_e$
   subject to : $\sum_{e: e \in IN(u)} x_e \leq 1$ for all $u \in V$
   $x_e \in \{0, 1\}$

   (a) Consider the LP relaxation $\mathcal{P}$ (in standard form) of this IP; that is, : maximize $\sum_{e \in E} x_e$
   subject to : $\sum_{e: e \in IN(u)} x_e \leq 1$ for all $u \in V$
   $x_e \leq 1$
   $x_e \geq 0$

   State the dual $\mathcal{D}$ of the primal $\mathcal{P}$ using dual variables y_u for $u \in V$. Can you explain this dual as the relaxation of a known optimization problem?

(b) Suppose now that we restrict attention to bipartite graphs. Explain (from anything you already know without any IP/LP theory) why the value of the LP OPT equals the value of the IP OPT.

3. Consider the unweighted vertex cover problem. Suppose you have a polynomial time algorithm $\mathcal{A}$ to compute the *size* of an optimal vertex cover for some class $\mathcal{G}$ of graphs closed under removal of edges and nodes (e.g. bipartite graphs). Show how to use $\mathcal{A}$ to compute an optimal *solution* (i.e. a subset of the vertices) for the vertex cover problem restricted to graphs in $\mathcal{G}$.

4. Use any of the above questions to argue why computing an optimal vertex cover in bipartite graphs can be done in polynomial time.

5. Consider the following *partial vertex cover problem*:

   Given a graph $G = (V, E)$ with vertex weights $w : V \to \Re^+$ and edge penalties $p : E \to \Re^+$.

   The goal is to select a subset of vertices $V' \subseteq V$ so as to minimize the sum of vertex weights in V' plus the sum of edge penalties for edges not covered by the vertices in V'.

   Express the problem as an IP and then use LP relaxation and rounding to derive an approximation algorithm. State the approximation ratio and provide an argument justifying why your IP defines the problem and why your algorithm achieves the approximation bound you are claiming.

6. BONUS QUESTION) Consider the following *doubly satisfied Max-Sat problem*:

   Given a propositional CNF formula $F = C_1 \wedge C_2 \ldots \wedge C_m$ with clause weights $w : \{C_1, \ldots, C_m\} \rightarrow \Re^+$.

   The goal is to find a truth assignment τ so as to maximize the expected weighted sum of clauses that are satisfied by at least two literals.

   Can you obtain a "good" approximation (beating the naive 1/4 approximation) for this problem using any approach? David Liu provided an example that shows that for the extension I had in mind, the analysis fails although the rounding actually gives an optimal result.

7. Consider the following slightly modified version of Schöning's algorithm where now the random walk only proceeds for n steps:

Choose a random assignment τ
Repeat n times   % n = number of variables
If τ satisfies F then stop and accept
Else Else Let C be an arbitrary unsatisfied clause
Randomly pick and flip one of the literals in C
End If

   (a) Show that this algorithm will succeed with probability at least $\frac{1}{2^n}(\frac{k+1}{k})^n$

   (b) Explain briefly why Schöning's algorithm repeats the random walk for $3n$ steps.

   (c) Suppose that we now allow each clause to be any (at most) k variable Boolean constraint. Explain whether or not the analysis in Schöning's algorithm still applies yielding the same bound on the expected running time.