CSC373F **Problem Set 4** **Fall, 2005**

Due: Friday, Dec 2 , beginning of lecture
Note change of due date by one week. Questions 2 and 5 have now become bonus questions.

NOTE: Each problem set only counts 5% of your mark, but it is important to do your own work (but see below). Similar questions will appear on the first term test which will cover material relating to both assignment 1 and assignment 2. You may consult with others concerning the general approach for solving problems on assignments, but you must write up all solutions entirely on your own. the University's Code of Behavior. You will receive 1/5 points for any question/subquestion for which you say "I do not know how to answer this question". You will receive .5/5 points if you just leave the question blank.

1. Show how to formulate the following problems as $\{0,1\}$ integer programming (IP) problems

   (a) The MaxSat problem where the input is a $F = C_1 \wedge C_2 \ldots C_r$ where each clause is a disjunction of literals.

   Hint: If there are r clauses and n propositional variables occuring in F, then it suffices to have $r+n$ variables occuring in the IP. [10 points]

   (b) The scheduling problem of maximizing the profit of scheduled jobs where each job $J_i = (p_i, d_i, v_i)$ and p_i is the processing time, d_i the deadline, and v_i the profit of job J_i. The jobs must be scheduled on one machine without overlap and every scheduled job must complete by its deadline.

   Hint: A subset of jobs can be feasibly scheduled if and only if they can be scheduled in order of their deadlines.

   [10 points]

2. Consider again the problem of maximizing the number of clauses satisfied by a CNF formula F. Now assume that there are at most two literals per clause. Consider the following local-search algorithm:

   Set $\tau(x_i) = true$ for all variables x_i.
   While there is a variable x_i for which complementing x_i will result in an increase in the number of clauses satisfied
   Complement $\tau(x_i)$
   End While

   We want to show that the local optimum τ (produced by the local search algorithm) is a 2/3 approximation (or 3/2 if you like approximation ratios to be ≥ 1). In fact, show that the number of clauses satisfied by τ is at least $(2/3) * r$ where r is the number of clauses in F.

   Hint: For each i, consider the number N_i^0 (respectively N_i^1) of clauses C_j such that no literal in C_j is satisfied by $\tau(x_i)$ (respectively 1 literal is satisfied and that literal is the one satisfied by $\tau(x_i)$. Argue that $N_i^0 \leq N_i^1$ for all i and then sum for all i.

   [20 points]

3. Consider the following CNF formula:
$F = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2) \wedge (\overline{x_3}) \wedge (x_3 \vee \overline{x_1})$

   (a) Consider the randomized algorithm that assigns truth values to the variables x_1, x_2, x_3 uniformly at random. What is the expected number of clauses in F that will be satisfied by this algorithm? [5 points]

   (b) Use the method of conditional expectations (say considering the variables in the order x_1, x_2, x_3, x_4) to derandomize this algorithm. What are the clauses in F that are now satisfied? [5 points]

4. Consider the following *dominating set problem*: We are given a collection of sets $\mathcal{T} = \{T_1, \ldots, T_m\}$ with $T_i \subseteq U$ for some universe U. There is also a cost function $c : U \rightarrow \Re^{\geq 0}$ and we let c_u denote the cost of element $u \in U$. A feasible solution is a subset $S \subseteq U$ such that $S \cap T_i \neq \emptyset$ for all i. The goal is to find a feasible subset S so as to minimize the cost $c(S) = \sum_{u \in S} c_u$.

   (a) Formulate the dominating set problem as a $\{0, 1\}$ IP [10 points]

   (b) Show how to use LP relaxation + rounding to obtain an d-approximation algorithm in the case that $|T_i| \leq d$ for every set T_i in the collection $\mathcal{T}$. [10 points]

5. (a) Consider the min capacity global cut problem where every edge e has an non-negative integer capacity $c(e)$. Explain why (as in the unweighted case) there are at most $\binom{n}{2}$ different min capacity cuts.

   [10 points]

   (b) Now consider the min cardinality $s - t$ cut problem on an unweighted graph G and the modified Karger randomized algorithm; that is, continue to contract random edges until there are two super nodes but never choose to contract an edge between the super node containing s and the super node containing t. Show that there is a class of *simple* graphs (i.e. no parallel edges) for which this algorithm will have an exponentially small (in n, the number of nodes in the graph) probability of finding the min cardinality $s - t$ cut.

   [10 points]