COMP 360 - Fall 2015 - Assignment 3

Due: 6:00 pm Nov 10th.

General rules: In solving these questions you may collaborate with other students but each student has to write his/her own solution. There are in total 105 points, but your grade will be considered out of 100. You should drop your solutions in the assignment drop-off box located in the Trottier Building.

---

1. (10 Points) Show that if we strengthen linear programming by also allowing constraints of the form $\sum_{i,j=1}^{n} a_{ij}x_ix_j = b$ (for integers b and a_{ij}), then the problem becomes NP-complete.

   Solution I: A feasible solution can be used as a certificate and can be verified easily. To show that the problem is NP-complete, we reduce the VertexCover problem to this problem. Since $y^2 = 1$ and $x = \frac{y+1}{2}$ is equivalent to saying that $x \in \{0, 1\}$, we can solve the VertexCover problem using the following program:

$$\begin{array}{ll} \min & \sum_u x_u \\ \text{s.t.} & x_u + x_v \leq 1 \quad \forall uv \in E \\ & y_u^2 = 1 \qquad\quad\; \forall u \in V \\ & x_u - \frac{y_u}{2} = \frac{1}{2} \quad \forall u \in V \end{array}$$

   Solution II: A feasible solution can be used as a certificate and can be verified easily. To show that the problem is NP-complete, we reduce the VertexCover problem to this problem. Note that the size of the minimum vertex cover is $\frac{K+n}{2}$ if K is the solution to the following program, and n is the number of vertices of G:

$$\begin{array}{ll} \min & \sum_u x_u \\ \text{s.t.} & x_u + x_v \leq 1 \quad \forall uv \in E \\ & x_u^2 = 1 \qquad\quad\; \forall u \in V \end{array}$$

   Indeed $x_u^2 = 1$ guarantees that $x_u \in \{-1, 1\}$, and the constraints $x_u + x_v \leq 1$ guarantee that the set S of the vertices that receive 1 form a vertex cover. Thus $K = \sum_u x_u = \sum_{u \in S} 1 + \sum_{u \notin S}(-1) = 2|S| - n$.

2. (10 Points) Show that if we strengthen linear programming by also allowing constraints of the form $|\sum_{i=1}^{n} a_i x_i| \geq b$ (for integers b and a_i), then the problem becomes NP-complete.

   Solution: Note that $|y| \geq 1$ and $y \leq 1$ and $-y \leq 1$ together imply that $y \in \{-1, 1\}$. Hence in the solution to the previous question, we can replace $y^2 = 1$ with these three constraints.

3. For each one of the following problems either prove that they are NP-complete or prove that they belong to P.

   (a) (10 Points)

   - Input: A CNF ϕ and a positive integer M.
   - Question: Is there a truth assignment that satisfies ϕ and assigns True to exactly M variables.

Solution: This problem is NP-complete. A truth assignment that satisfies ϕ and assigns True to exactly M variables can be used as a certificate and can be verified efficiently for a YES input.

To prove the completeness we reduce SAT to this problem. Given input ϕ to SAT (with n variables) it suffices to run the oracle for the above problem for $M = 0, \ldots, n$. If the oracle outputs YES on any of these inputs then ϕ is satisfiable, if it outputs NO on all the values of M, then ϕ is not satisfiable.

(b) (15 Points)

- Input: Positive integers $a_1, \ldots, a_n$ and a positive integer M.
- Question: Is there a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} a_i \in \{M-1, M, M+1\}$?

Solution: This problem is NP-complete. A subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} a_i \in \{M-1, M, M+1\}$ can be used as a certificate and can be verified efficiently for a YES input.

To prove the completeness we reduce SUBSETSUM to this problem. Given an input $\langle w_1, \ldots, w_n, M\rangle$ to the SUBSETSUM problem, we create the input $\langle 3w_1, \ldots, 3w_n, 3M\rangle$ for the above problem. Note that $\sum_{i \in S} 3w_i \in \{3M-1, 3M, 3M+1\}$ if and only if $\sum_{i \in S} w_i = M$. Hence we can run the oracle on $\langle 3w_1, \ldots, 3w_n, 3M\rangle$ and if it outputs YES then the answer to the SUBSETSUM input is YES and otherwise it is NO.

(c) (15 Points)

- Input: A CNF ϕ.
- Question: Is there a truth assignment that satisfies none of the clauses in ϕ.

Solution I: This problem is in P. Note that if a variable x_i and its negation $\overline{x_i}$ both appear in the formula then the answer is NO (as in every assignment one of them will be TRUE). Otherwise every variable appears either as x_i or in the negated form $\overline{x_i}$ (but not both). We can decide the value of x_i accordingly ($x_i = F$ if x_i appears, and $x_i = T$ if $\overline{x_i}$ appears) , and this will not satisfy any clauses. Hence the following algorithm solves the problem: Check to see whether there is a variable x_i such that both x_i and $\overline{x_i}$ appear in the formula. In this case output NO, otherwise output YES.

Solution II: This problem is in P. Note that every clause will uniquely determine the value of all the variables that are involved in that clause (since all the terms in the clause must be false). Hence we can start from the first clause and set the values of the variables accordingly. If at any point we reach a term that has already been set to TRUE, then we terminate and output NO. Otherwise when all the clauses are processed we output YES.

(d) (15 Points)

- Input: A graph G and a positive integer M.
- Question: Does G have a proper 2-coloring with colors R, G such that exactly M vertices receive the color R?

Solution: This problem is NP-complete. A proper 2-coloring with colors R, G such that exactly M vertices receive the color R can be used as a certificate and can be verified efficiently for a YES input.

To prove the completeness we reduce SUBSETSUM to this problem. Given an input $\langle w_1, \ldots, w_n, K\rangle$ to the SUBSETSUM problem, we create a graph that is a disjoint union of n stars with number of vertices $w_1 + 2, \ldots, w_n + 2$ respectively (the i-th star consists of a vertex connected to $w_i + 1$ leaves). We also set $M = n + K$. Note that every star can be colored in two ways, either the center is colored R and the leaves are colored B, or vice versa. Let S be the set of all i, such that in the i-th star, the leaves are colored R. Then the total number of red vertices will be $\sum_{i\in S}(w_i+1)+\sum_{i\notin S}1 = n+\sum_{i\in S} w_i$. Note that this number is equal to M if and only if $\sum_{i\in S} w_i = K$.

Hence we can run the oracle on this graph with $M = n+K$, if it outputs YES, then the answer to the SUBSETSUM problem is YES, and otherwise it is NO.

(e) (15 Points)

- Input: A graph G.
- Question: Does G have a proper 3-coloring with colors R, G, B such that at most 100 vertices receive the color R?

Solution: This problem is in P. Let n denote the number of the vertices of G. There are at most n^{100} different ways of choosing at most 100 vertices from G. For each one, we color these at most 100 vertices R, and then try to color every other vertex with two colors G and B (this can be done in polynomial time $O(n^2)$). Hence the total running time is at most $O(n^{102})$.

(f) (15 Points) Recall that a graph is called Hamiltonian if it contains a cycle that visits all the vertices.

- Input: A graph G.
- Question: Is G a Hamiltonian *bipartite* graph?

Solution: This problem is NP-complete. A Hamiltonian cycle can be used as a certificate for a YES input and whether it is a Hamiltonian cycle and whether the graph is bipartite can be verified in polynomial time easily.

To prove the completeness we reduce the Hamiltonian cycle problem to this problem. Consider an input G for the Hamiltonian cycle problem. For every vertex v of G, we add four vertices v^-, v_+, v_-, v^+ together with the path $v^-, v_+ v_- v^+$ to the new graph G'. Furthermore if uv is an edge in the original graph G, then we add the edges $u^- v^+$ and $v^+ u^-$ to G'. Note that a Hamiltonian cycle in the original graph can be easily translated to a Hamiltonian cycle in G'. Furthermore consider a Hamintonian cycle in G'. Since the only way to visit v_+ is through the path $v^- v_+ v_- v^+$ (and a Hamiltonian cycle has to visit all the vertices), these three vertices must appear together on the cycle (either as $v^- v_+ v_- v^+$ or as $v^+ v_- v_+ v^-$). As a result any Hamiltonian cycle in G' will correspond to a Hamiltonian cycle in the original graph G. Hence G has a Hamiltonian cycle if and only if G' has a Hamiltonian cycle. Moreover note that G' is bipartite. Hence we can run the oracle on G' to see whether G is Hamiltonian or not.