Introduction to Algorithms December 11, 2003
Massachusetts Institute of Technology 6.046J/18.410J
Professors Shafi Goldwasser and Silvio Micali Handout 33

Problem Set 9 Solutions

Problem 9-1. The challenges of industrial research

(a) First let us run DECISION-SG(G, d). If it outputs 0, then we are done since that means that G does not have a degree d spanning graph.

Let us arbitrarily order the m edges in E: $e_1, \ldots, e_m$, and run the following algorithm:

SEARCH-SG
$E' \leftarrow E$
for $i \leftarrow 1$ **to** m
  do
  $E' \leftarrow E' - e_i$
  if DECISION-SG$(< (V, E'), d >) = 0$
    then $E' \leftarrow E' + e_i$
return E'

We need to show that the resulting graph $G' = (V, E')$ is a degree d spanning graph of G: i.e. G' is connected and no vertex has degree greater than d in G'.

To prove that G' is connected, observe that after each iteration of the loop, the edge set E' is such that DECISION-SG$(< (V, E'), d >)$ accepts, because if the removal of an edge e_i from E' causes DECISION-SG$(< (V, E'), d >)$ to reject, then e_i is brought back into E'. By definition of a degree d spanning graph, it follows that the graph G' is connected.

It remains to be shown that no vertex has degree greater than d. We do so by contradiction. Suppose not all vertices of G' have degree at most d. Let $G'' = (V, E'')$ be a degree d spanning graph of G'. (By construction, we know that it exists.) Some vertex v of G' has degree greater than d in G', but at most d in G''. Then there is an edge $e \in E'$ adjacent to v such that $e \notin E''$. But then DECISION-SG must have returned 1 when our algorithm tried to delete e from E', and so $e \notin E'$. This is a contradiction.

Note: This is an NP-complete problem known as degree constrained spanning tree, discussed on p. 206 of Garey and Johnson's *Computers and Intractability*.

(b) Recall that 3-SAT is NP-complete. Therefore, if the magic box solves it in polynomial time, any problem in NP has a polynomial-time algorithm.

Now let us show that the decision problem DECISION-SG is in NP. All that is needed to prove that $G = (V, E)$ has a degree d spanning graph is to produce a graph $G' = (V, E')$ which is a degree d spanning graph of G. The size of the encoding of G' is at most linear in the size of the encoding of G. In order to verify that G' is a degree d spanning subgraph, all that needs to be done is to make sure that $E' \subseteq E$ and $\forall v \in V$,

$\deg(v) \leq d$. Thus this problem has a polynomial-size certificate and a polynomial-size verification procedure, which implies that it is in NP.

Therefore it follows that a magic box that solved DECISION-SG in polynomial time can be constructed. By part (a), it follows that SEARCH-SG can be solved in polynomial time.

Problem 9-2. Approximating knapsack

(a) Just make $w_i = s_i$ for all i.

(b) Suppose we have $w_1 = W, s_1 = B, w_2 = 2W/B, s_2 = 1$. So clearly our algorithm only takes object 2, for value $2W/B$, when we could have had value W. So we get only a $2/B$ fraction of the optimal total worth. B can be arbitrarily large, so this can be arbitrarily bad.

(c) First, let's discard every item with $s_i > B$, because they are irrelevant. Now, let i be the first object the greedy algorithm rejects. (If no item is ever rejected then we obviously get an optimal solution.) If at this time, the backpack were exactly full, we'd obviously be optimal, because we'd have filled it with the highest value/size objects (that is replacing any objects with others of the same size could only lower the value). So suppose it isn't. But if we had a slightly bigger backpack, such that item i did exactly fit, then we'd have an optimal solution for the bigger backpack. A solution for a bigger backpack can be no worse, so the optimal value can be no bigger that what greedy has taken so far plus the value of object i. Thus the larger of what greedy has taken so far and w_i is at least half of optimal. (Rephrased, optimal is no more than the sum of the two, so the bigger of the two is at least half.) What greedy finally takes is only bigger than what it has so far, and the highest value object is only higher than w_i, so the better of these must also be with a factor of 2 of optimal.

Problem 9-3. Reductions

(a) If $P = NP$, then this is true. Otherwise, this is false: L_2 can be NP-complete. Therefore this is an open question.

(b) True.

(c) True. Proof by contradiction: suppose L_1 is NP-complete. Since $L_1 <_p L_2$, L_2 is NP-complete by transitivity of reductions. This is a conradiction.

(d) If $P = NP$, then this is true, since then any problem in P is NP-complete. Otherwise, it is false: we can take $L_1, L_2 \in P$, and they are not NP-complete. Therefore, this is an open problem.

(e) True by definition of NP-completeness.

Problem 9-4. Easy vs. hard cases of the same problem

(a) It is clear that this is a problem in NP, since in order to prove that a digraph has a Hamiltonian path, it is sufficient to provide an ordering of the vertices; and to verify that a given ordering of the vertices represents a Hamiltonian path, it is sufficient to check that any two consecutive vertices v_i, v_{i+1} are joined by an edge (v_i, v_{i+1}).

We now need to show that this is an NP-hard problem.

First, reduce Hamiltonian cycle to directed Hamiltonian cycle: suppose we are given an undirected graph $G = (V, E)$. Create a directed graph $G' = (V, E')$, where $(u, v), (v, u) \in E'$ if $(u, v) \in E$. If this new graph has a directed Hamiltonian cycle, then the original graph G must have a Hamiltonian cycle, and the other way around.

Then, reduce directed Hamiltonian cycle to directed Hamiltonian path: suppose we are given a digraph $G = (V, E)$. Construct a new graph $G' = (V', E')$ as follows: Pick and arbitrary vertex $v \in V$ and split it into two vertices: v_0 and v_1. Let $V' = (V - v) \cup \{v_0, v_1\}$. For all edges $(v, u) \in E$, add an edge $(v_0, u) \in E'$. For all edges $(u, v) \in E$, add an edge $(u, v_1) \in E'$. For all other edges in E, copy them to E'.

We claim that G' has a Hamiltonian path from v_0 to v_1 if and only if G has a directed Hamiltonian cycle.

Suppose G has a Hamiltonian cycle. Then it is clear from the construction that G' will have a Hamiltonian path from v_0 to v_1.

Suppose G' has a Hamiltonian path from v_0 to v_1. Then there is a sequence of edges in E' that start at v_0 and end in v_1 and visit every node exactly once. From the way E' was constructed, it follows that there exists a path in G that starts at v and ends at v such that it visits every node $u \neq v$ exactly once and v exactly twice. Such a path in G is its Hamiltonian cycle.

Therefore it follows that the directed Hamiltonian path problem is NP-complete.

(b) Let us do a topological sort on the vertices of G and consider the resulting ordering of the vertices. A Hamiltonian path from u to v exists if and only if u is the first vertex, v is the last vertex, and if v_i is followed by v_j in this ordering, then there is an edge from v_i to v_j.