Analysis of Algorithms - Home Work 5

CSE 548 Fall '19

Prof. Jie Gao

Submission By:

Narayan Acharya

112734365

Contents

1 Question 1

Textbook [Kleinberg & Tardos] Chapter 8, page 505, problem #14.
Solution:

We reduce well known problem NP-Hard problem, Independent Set (IS), to our problem at hand Multiple Interval Scheduling (MIS). We ensure this reduction is in polynomial time.

Consider the following reduction, for every vertex v_i in IS we construct a job in MIS, say x_i and for every edge e_{ij} in IS we construct an interval in MIS, y_{ij} such that vertices v_i and v_j require the interval y_{ij} for their job.

Now, if we wan find an a schedule that satisfies our constraints, i.e. we can find k non-conflicting job from our n jobs, then we can prove that our IS graph, say G, has an independent set of size k.

Now, consider the 2 scenarios:

1. If we assume G contains an independent set of vertices, v_{i1}, v_{i2}, ..., v_{ik} of size k then it follows that the jobs x_{i1}, x_{i2}, ..., x_{ik} are non-conflicting since the corresponding vertices will not be sharing any edges which results in the jobs not sharing any intervals.

2. To prove the same in the other direction, If we assume that we have k non-conflicting jobs x_{i1}, x_{i2}, ..., x_{ik} then the corresponding vertices in G are independent, i.e. v_{i1}, v_{i2}, ..., v_{ik} are an independent set of of size k. If they are not independent then there must be some vertices, say v_{im} and v_{in} that have an edge e_{imin}. But this would imply that the jobs x_im and x_in share an interval. This contradicts our initial assumption that the jobs our non-conflicting.

The above 2 scenarios proves that solving for MIS would solve for IS. Thus, MIS is NP-Complete.

2 Question 2

Textbook [Kleinberg & Tardos] Chapter 8, page 505, problem #16.
Solution:

We reduce 3-Dimensional Mapping (3DM) to the Intersection Inference problem (II). Constraints for 3DM are defined on sets X, Y, Z, T such that $X \cap Y = Y \cap Z = X \cap Z = \emptyset$, $|X| = |Y| = |Z| = n$ and $T \subseteq X \times Y \times Z$.

We reduce as follows: let $U = T$ and let $A_1, A-2, ...A_n$ be the subset of triplets that contain $x_1, x_2, ...x_n \in X$ respectively. to be the subsets of triples that contain respectively. We do this for $y_i \in Y$ from $y_{n+1}...y_{2n}$ and $z_i in Z$ from $z_{2n+1}...z_{3n}$. Also $c_0 = n$ and $c_i = 1 \forall i \in 1, 2..n$. This can be done in polynomial time.

This transformation restricts for n triples must match, and each triple contains a specific element from $X \cup Y \cup Z$. This means constraints for 3DM were met. If a solution could not be found then some constraint could not be met, either elements were covered with duplication or some element(s) was/were missed. This implies that if we have solved for II, we have solved for 3DM.

This means we were successfully able to reduce 3DM to II and given 3DM is NP-Complete, II too is NP-Complete.

3 Question 3

Textbook [Kleinberg & Tardos] Chapter 8, page 505, problem #17.
Solution:

We reduce well known problem Subset-Sum (SS) Problem, to our problem at hand Zero Weight Cycle (ZWC) Problem. We ensure this reduction is in polynomial time.

We construct an instance of ZWC from SS as follows: We construct a directed graph of $2n$ nodes, 2 nodes for each number in our set from SS. For every number x_i we add 2 nodes, v_i and u_i to our graph. We had a directed edge from v_i to u_i with weight the value of x_i. We also add an edge from every vertex u_j such that $j \neq i$ to v_i with edge weight 0. We do this the other way round as well, that is edges from v_j to u_i, where $j \neq i$, with edge weights 0. We can do this in polynomial time.

Now, we know that based on our construction, we can have cycles where we alternate between nodes from v and u. The weight of the cycle will be 0 if and only if the sum of all the weights between u_i and v_i is 0. This would require the sum of all corresponding a_i to be 0 which implies we have found a subset of numbers with sum 0.

If we were able to solve for ZWC implies we can solve for SS with polynomial reduction. But given SS is known to be NP-Complete we can claim that ZWC is NP-Complete.

4 Question 4

Textbook [Kleinberg & Tardos] Chapter 8, page 505, problem #20.
Solution:

We reduce k-Coloring problem which is known to be NP-Complete to our problem at hand, Low Diameter Clustering Problem (LDC) in polynomial time.

For out k-Coloring problem, consider a graph, $G = (V, E)$ and a number k. We are constrained such that, coloring, say a function c is achieved such that $\forall(v, u) \in E \Rightarrow c(v) \neq c(v)$. We reduce as follows: For every vertex, v in V we have a corresponding object in p_v in LDC. We define our distance function for LDC where a is some constant, $a > 1$, and B is our boundary distance -

1. $\forall(v, u) \in E \Rightarrow d(p_v, p_u) = aB$
2. $\forall(v, u) \notin E \Rightarrow d(p_v, p_u) = B/a$
3. $\forall(v, u), v = u \Rightarrow d(p_v, p_u) = 0$

Now, consider the 2 forward and backward scenarios:

1. If we have a solution for LDC then we have solved for k-Coloring as well. For any 2 objects, p_u and p_v where $d(p_v, p_u) > B$, we know that there will not be an edge in E, i.e. $(u, v) \notin E$. This implies u and v can have the same color. Now, we can color the subset of vertices that satisfy this condition with a unique color and do this for each subset by coloring them with a color not chosen previously.
2. For the other direction, if we have solved for k-Coloring then we have solved for LDC as well. Consider this, if $c(u) \neq c(v)$ for any vertices v and u in V for $(v, u) \in E$ implies $d(p_u, p_v) > B$ and p_u and p_v cannot be in the same cluster. We can cluster the objects such that for every unique color $c_1, c_2, ...c_k$ we create subsets from P, say $s_1, s_2...s_k$ and assign the same color c_i to all objects in subset s_i

This implies LDC is NP-Complete.

5 Question 5

Textbook [Kleinberg & Tardos] Chapter 11, page 651, problem #9.
Solution:

Narayan Acharya
112734365

Consider the following variables and constraints for our 3 Dimensional Matching - sets X, Y, Z, T such that $X \cap Y = Y \cap Z = X \cap Z = \emptyset$, $|X| = |Y| = |Z| = n$ and $T \subseteq X \times Y \times Z$.

We use the simple strategy of adding triplets to our result set, say R, if does not conflict with already existing triplets in R. This restricts an element belonging to X, Y or Z to be part of at most one triplet. Given this restriction, it is easy to prove the size of R will be a third of the optimal matching, say S.

For a triple our optimal matching result, S, must intersect with at least one triplet from R else we could add this triple to R to extend it further. Given, X, Y and Z contain no common elements, each triplet from R can at maximum have 3 conflicts with triplets from S. Therefore, size of S can be at max three times that of our approximate solution R.

6 Question 6

Textbook [Kleinberg & Tardos] Chapter 11, page 651, problem #10.
Solution:

a If $v \in S$ that means at some stage of the given greedy algorithm v was chosen and made it to the result S. If $v \notin S$ implies that at some stage of the algorithm there was another node, v', which was picked instead of v. If v' was a neighbor to v, i.e. (v, v') is an edge in G, v would automatically dropped based on our algorithm and never make it to S. The only reason for v' to be chosen over v would be that $w(v') \geq w(v)$.

b We use a charging scheme to prove this. Consider there exists another independent set R. For every node, say v, in R we have the following 2 possibilities:

   (a) If $v \in S$ then we charge v to itself. Given S and R are independent sets, no other node is charged to v.

   (b) If $v \notin S$ implies that some other node v', a neighbor of v, was selected instead of v and we charge v to v'. In this case, no more than 4 nodes, i.e. its neighbors are charge to v. The total weight of these 4 nodes cannot exceed 4 times the weight of v' (worst possible scenario is all nodes under consideration, v' and its 4 neighbors, are of same weight). These charges reflect in the total weight of R we can say that weight of R is at most 4 times the weight of S.

References

[1] Amanpreet Singh,
`Discussed approaches with Amanpreet Singh (SBU ID - 112044129)`