CSCI3160: Regular Exercise Set 12

Prepared by Yufei Tao

Problem 1 (Reduction from Vertex Cover to Set Cover). Prove: If the set cover problem admits a polynomial time algorithm (for finding an optimal solution), then the vertex cover algorithm admits a polynomial time algorithm (for finding an optimal solution).

Solution. Let $G = (V, E)$ be the input graph to the vertex cover problem. We will create an instance of the set cover problem such that if we can solve that instance optimally, we can derive an optimal solution for the vertex cover problem on G efficiently.

For each vertex $u \in V$, create a set S_u that includes all the edges of E incident on u. We will call u the *owner* of S_u. Now, define a collection of sets:

$$\mathcal{S} = \{S_u \mid u \in V\}.$$

The universe U equals $\bigcup_{u \in V} S_u = E$. For any sub-collection $\mathcal{C} \subseteq \mathcal{S}$, define

$$V(\mathcal{C}) = \{u \in V \mid S_u \in \mathcal{C}\}$$

namely, $V(\mathcal{C})$ includes all the owners of the sets in $\mathcal{C}$.

> **Claim 1:** $\mathcal{C} \subseteq \mathcal{S}$ is a set cover of U (namely, $\bigcup_{S \in \mathcal{C}} S = U$) if and only if $V(\mathcal{C})$ is a vertex cover of G (namely, every edge of G has at least a vertex in $V(\mathcal{C})$).

Proof of Claim 1: First, assuming that $\mathcal{C} \subseteq \mathcal{S}$ is a set cover of U, we will prove that $V(\mathcal{C})$ is a vertex cover of G. For this purpose, we must prove that any edge $e = \{u, v\}$ of G must have at least one vertex in $V(\mathcal{C})$. As e is a member of U, it must be covered by $\bigcup_{S \in \mathcal{C}} S$ (as $\mathcal{C}$ is a set cover), which means e must appear in at least one set $S \in \mathcal{C}$. Note that the owner of S must be either u or v (by the way we construct the sets in $\mathcal{S}$). W.l.o.g., suppose that u is the owner of S, which implies $u \in V(\mathcal{C})$. Therefore, $V(\mathcal{C})$ is a vertex cover of G.

Second, assuming that $V(\mathcal{C})$ is a vertex cover of G, we will prove that $\mathcal{C}$ is a set cover of U. For this purpose, we must prove that any edge $e = \{u, v\}$ of G must be covered by $\bigcup_{S \in \mathcal{C}} S$. As $V(\mathcal{C})$ is a vertex cover, it must contain either u or v. W.l.o.g., suppose that $u \in V(\mathcal{C})$. By definition of $V(\mathcal{C})$, u must be the owner of a set in $\mathcal{C}$ (the set is S_u). It thus follows that $e \in S_u \subseteq \bigcup_{S \in \mathcal{C}} S$. $\square$

> **Claim 2:** For any subset $U \subseteq V$, there is a sub-collection $\mathcal{C} \subseteq \mathcal{S}$ satisfying $V(\mathcal{C}) = U$.

Proof of Claim 2: Such a $\mathcal{C}$ can be explicitly constructed as $\mathcal{C} = \{S_u \mid u \in U\}$. $\square$

Therefore, to solve the vertex cover optimally, we can first find an optimal set cover $\mathcal{C}$ and then return $V(\mathcal{C})$. Hence, if $\mathcal{C}$ can be found in polynomial time, we obtain a polynomial time algorithm for the vertex cover problem.

Remark. This means that if the vertex cover cannot be solved in polynomial time, neither can set cover.

Problem 2. Let $\mathcal{C}^*$ be an optimal universe cover for the set cover problem. Consider running the set cover algorithm discussed in the lecture. In particular, consider the moment right before the algorithm is to choose the i-th set S_i, having chosen already $S_1, S_2, ..., S_{i-1}$. Let z_{i-1} be the number of elements in the universe that have not been covered by $S_1 \cup S_2 \cup ... \cup S_{i-1}$. Let $\mathcal{C} = \{S_1, S_2, ..., S_{i-1}\} \cap \mathcal{C}^*$, i.e., $\mathcal{C}$ includes all the sets of $\mathcal{C}^*$ that "happened" to have been selected by our algorithm so far. Prove:

- $\mathcal{C} \neq \mathcal{C}^*$.
- S_i has benefit at least $z_{i-1}/(|\mathcal{C}^*| - |\mathcal{C}|)$.

Solution.

(i) If $\mathcal{C} = \mathcal{C}^*$, then the sets picked by our algorithm have already covered the entire universe. In that case, the algorithm would not have needed to pick S_i.

(ii) Let $\mathcal{C}^*_{out} = \mathcal{C}^* \setminus \mathcal{C}$; note that $|\mathcal{C}^*_{out}| = |\mathcal{C}^*| - |\mathcal{C}|$. The z_{i-1} elements not yet covered by $\{S_1, S_2, ..., S_{i-1}\}$ must be covered by $\mathcal{C}^*_{out}$. Therefore, one of the sets in $\mathcal{C}^*_{out}$ must have a benefit at least $z_{i-1}/|\mathcal{C}^*_{out}| = z_{i-1}/(|\mathcal{C}^*| - |\mathcal{C}|)$. The claim thus follows from the algorithm's greedy nature.

Problem 3. Let $\mathcal{I}$ be a set of n intervals in 1D space (i.e., each interval has the form $[x, y]$) and P be a set of n 1D points. A subset $S \subseteq P$ is a *stabbing set of* $\mathcal{I}$ if every interval of $\mathcal{I}$ covers at least one point in S. Let OPT be the size of the smallest stabbing set of $\mathcal{I}$, and you are guaranteed that OPT ≥ 1. Design an algorithm to find a stabbing set of size at most OPT $\cdot\, O(\log n)$. Your algorithm must run in time polynomial to n.

Solution. We will convert the problem into an instance of set cover. For each point $p \in P$, define I_p to be the set of intervals $I \in \mathcal{I}$ that contain p. The set collection $\{I_p \mid p \in P\}$ defines a set cover problem with $U = \bigcup_{p \in P} I_p$. A subset $S \subseteq P$ is a stabbing set of $\mathcal{I}$ if and only if $\{I_p \mid p \in S\}$ is a set cover of U. By running our greedy set-cover algorithm, we can obtain a set cover $\mathcal{C}$ of U with size OPT $\cdot\, O(\log n)$. We can then return $\{p \in P \mid I_p \in \mathcal{C}\}$ as a stabbing set for the original problem.

Problem 4*. Let $\mathcal{I}$ be a set of n intervals in 1D space (i.e., each interval has the form $[x, y]$). A set S of 1D points is a *stabbing set of* $\mathcal{I}$ if every interval of $\mathcal{I}$ covers at least one point in S. Let OPT be the size of the smallest stabbing set of $\mathcal{I}$. Design an algorithm to find a stabbing set of size at most OPT $\cdot\, O(\log n)$. Your algorithm must run in time polynomial to n.

Remark. The problem is similar to Problem 3 except that we can form a stabbing set with arbitrary 1D points, rather than using only points from a given set P.

Solution. Let P be the set of $2n$ endpoints of the intervals in $\mathcal{I}$. Convince yourself that it suffices to form stabbing sets from the points in P. Now, we can apply the solution to Problem 3.