CS218 Design and analysis of algorithms **Jan-Apr 2023**

Exercises: NP, NP-completeness, Randomized algorithms (No submission)

Independent set: given a graph and a number k, is there a set of k vertices such that there is no edge between them?

Vertex Cover: given a graph and a number k, is there a set S of k vertices which covers all edges (i.e., every edge in the graph has at least one end point in S).

Set Cover: Suppose we are given a collection of subsets of a set U, say $S_1, S_2, \ldots, S_r \subseteq U$. Can we select k of these subsets such that their union is U?

Directed Hamiltonian cycle: Given a directed graph with n vertices, is there a (directed) cycle of length n? By definition, a cycle cannot have repeated vertices.

1. Reduce the independent set problem to the vertex cover problem.

2. Reduce vertex cover problem to the set cover problem.

3. Suppose the following version of knapsack problem is NP-complete. Given a set of integer weights $w_1, w_2, \ldots, w_n$ and target weights W_1, W_2, is there a subset S of the weights whose sum is between W_1 and W_2, i.e., $W_1 \leq \sum_{i \in S} w_i \leq W_2$?

   Using this fact, prove that the following load balancing problem is NP-complete. Given a set of integer loads $t_1, t_2, \ldots, t_n$ and a target makespan T, is there a way to distribute all the loads to two machines so that the maximum load on any machine is at most T?

4. *Integer programming*: Given a set of linear inequalities in variables $x_1, x_2, , x_n$, decide if there is an integer solution satisfying all of them simultaneously. For example the set

$$\begin{aligned} 0 \leq x_1, x_2 &\leq 1 \\ x_1 - x_2 &\geq 0. \end{aligned}$$

   has an integer solution $(1, 0)$ (and also $(1, 1)$, $(0, 0)$).

   Show that Integer Programming is NP-hard. You can try a reduction from SAT to this problem. Given a CNF Booelan formula ϕ, you need to generate a set S of linear inequalities and prove that ϕ is satisfiable if and only if the set S has an integer solution.

5. Reduce SAT problem to Directed Hamiltonian cycle problem.

Randomized algorithms

1. Suppose you have designed a randomized algorithm count the number of distinct visitors to your website. Suppose the actual number of distinct visitors is d and your estimate is c. Suppose your algorithm gives the following guarantee:

$$\Pr[d/4 \leq c \leq 3d/4] \geq 0.75$$

   A common way to increase the confidence (probability of success) is run the same algorithm multiple times. Here we can run k copies of the same algorithm in parallel and finally output the median of all estimates. Let the median of all estimates be c^*. Use Chernoff bounds to prove that

$$\Pr[d/4 \leq c^* \leq 3d/4] \geq 1 - e^{-k/12}$$

   Chernoff bound: Let X be a sum of k random independent variables taking values in $\{0, 1\}$. Let $\mu = \mathbb{E}[X]$ be the expected value. Then

$$\Pr\left(X \geq (1 + \delta)\mu\right) \leq e^{-\delta^2 \mu/(2+\delta)}.$$

Approximation algorithms

1. Recall the algorithm that gives 2-approximation for minimum size vertex cover.

   While the graph is non-empty

       choose an edge (u, v) and put both its endpoints in S

       delete u and v and all their incident edges

       delete isolated vertices

   Observe that the edges chosen during the algorithm form a matching in the given graph. Prove that this is an 1/2-approximation for maximum matching. That is, the matching obtained has size at least half of the maximum size matching.

2. **Maximum weight matching:** Given a graph with edge weights, the goal is to find a matching (set of disjoint edges) with maximum total weight. Write an integer linear program for the maximum weight matching problem. Now, remove the integer constraint, that is, variables are allowed to take any real value. We get a linear program. Find an example (a graph with edge weights), where the optimal value of the linear program is higher than the weight of the maximum weight matching. Interestingly, if the graph is bipartite then the two values are always equal.

3. Suppose you want to promote an ad on twitter. You have identified a set of n influencers on twitter, each of which have a significant number of followers you are targeting. You have obtained the lists of their followers, let these lists be $L_1, L_2, \ldots, L_n$. Note that these lists may have many common followers among them. Let's say the influencers are somewhat diverse in the sense that any follower is present in **at most 3** out of these n lists. We want to select the minimum size subset of influencers, without compromising on their reach. That is, every follower in the union of the n lists is covered by at least one selected influencer. Design a 3-approximation algorithm for this problem.

4. Suppose each influencer is asking for a price for promoting your ad, let these prices be $p_1, p_2, \ldots, p_n$. Now, your goal is to select a subset of influencers with minimum total price such that every follower in the union of the n lists is covered by at least one selected influencer.

   We write the following LP.

$$\min \sum_{i=1}^{n} p_i x_i \text{ subject to}$$
$$1 \geq x_i \geq 0 \text{ for each } 1 \leq i \leq n$$
$$\sum_{i \text{ followed by } f} x_i \geq 1 \text{ for each follower } f$$

   The last summation is over all influencers i such that the follower f is in his/her list L_i. Recall the assumption that for any follower there are at most 3 such influencers. Suppose an LP solver gives us an optimal solution for this LP, which need not be an integer solution.

   - Give an appropriate rounding scheme so that the obtained set of influencers cover all followers.
   - Ensure that the price of the obtained set of influencers is at most 3 times the optimal price.