CS 170 HW 8 (Optional)

Due **2020-03-16, at 10:00 pm**
You may submit your solutions if you wish them to be graded, but they will be worth no points

1 DP solution writing guidelines

Try to follow the following 3-part template when writing your solutions.

- Define a function $f(\cdot)$ in words, including how many parameters are and what they mean, and tell us what inputs you feed into f to get the answer to your problem.
- Write the "base cases" along with a recurrence relation for f.
- Prove that the recurrence correctly solves the problem.
- Analyze the runtime and space complexity of your final DP algorithm? Can the bottom-up approach to DP improve the space complexity?

2 Motel Choosing

You are traveling along a long road, and you start at location $r_0 = 0$. Along this road, there are n motels at location $\{r_i\}_{i=1}^n$ with $0 < r_1 < r_2 < \cdots < r_n$. The only places you may stop are these motels, but you can choose which to stop at. You must stop at the final motel (at distance r_n), which is your destination.
Ideally, you want to travel exactly T miles a day and stop at a motel at the end of the day, but this may not be possible (depending on the spacing of the motels). Instead, you receive a *penalty* of $(T - x)^8$ each day, if you travel x miles during the day. The goal is to plan your stops to minimize the total penalty (over all travel days).
Describe and analyze an algorithm that outputs the minimum penalty, given the locations $\{r_i\}$ of the motels and the value of T.

Solution: Let $P[i]$ be the optimal (*i.e.* minimum) penalty for getting to hotel i; it holds that

$$P[i] = \min_{j<i} \left\{ P[j] + (T - (r_i - r_j))^8 \right\} .$$

The base case is given by $P[0] = 0$. We can compute the recurrence as follows.

Input: $\{r_i\}, T$
$P[0] := 0$
for $i = 1$ to n:
  $P[i] = 0$
  for $j = 1$ to $i - 1$:
    if $P[i] > P[j] + (T - (r_i - r_j))^8$
      $P[i] = P[j] + (T - (r_i - r_j))^8$
Output: $P[n]$

The running time is $O(n^2)$. To prove correctness, note that the base case is trivial. Assume as an inductive hypothesis that $P[j]$ are computed correctly for all $j < n$. Then to finally get to the destination at location r_n, there has to be a penultimate stop is i along our route. Then each choice of the stop i lead to a penalty of $P[i] + \left(T - (r_i - r_n)^8\right)$. We assumed that the first term is the minimum penalty to get to i, and the second term is the necessary penalty from i to n. Therefore, minimizing this quantity over all i leads to the minimum penalty of the entire trip.

3 Power of LP

In this problem, we are going to see many problem we have studied so far in the class can be expressed as a linear program (LP). For each problem, we ask you to first provide an *integer linear programming* (ILP) formulation. An integer linear program is just like a linear program (with linear constraints and objective), but allows you to add the constraints that the variables are integral. For example, you can have $x_e \in \{0, 1\}$— which is not allowed in LP.

The requirement is that the optimal solution to the ILP should correspond to an optimal solution to the original problem (for example, MST). For each problem, try to find an LP with the smallest number of constraints and variables. Some of them may require exponentially many constraints, but it's OK.

(a) Given a weighted, undirected graph, write an ILP formulation for the problem of minimum spanning tree, and describe how to relax it to an LP. Argue that any feasible solution to the ILP is a spanning tree, and that any spanning tree is a feasible solution to the ILP.

Solution: For every cut $\left(S, \overline{S}\right)$, we must select at least one edge that crosses the cut. For a subset $S \subset V$, let $\delta(S)$ be the edges crossing the cut (one endpoint in S, the other in $V - S$). Also, a spanning tree must contain $n - 1$ edges. This leads to the following ILP. (The $n - 1$ constraint is not technically necessary, so we give full credit to solutions without this constraint.)

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in E} w_e x_e \\ \text{subject to} \quad & \sum_{e \in E} x_e = n - 1 \\ & \sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S \subset V, S \neq \emptyset \\ & x_e \in \{0, 1\} \qquad \forall e \in E \end{aligned}$$

We claim that any feasible solution is a spanning tree, where the tree consists of all edges whose variables are set to 1. First, if a component S is isolated, then the cut constraint corresponding to S will be violated. Hence, any feasible solution is a connected graph. It follows that the constraint that $\sum_e x_e = n - 1$ ensures it is a tree.

An alternative solution is to consider that on any subgraph an spanning tree also has to span all vertices and remain a tree. We let $E(S)$ to denote all edges whose both endpoints lie in S, where $S \subseteq V$.

$$\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} w_e x_e \\
\text{subject to} \quad & \sum_{e \in E} x_e = n - 1, \quad \forall S \subset V, S \neq \emptyset, V \\
& \sum_{e \in E(S)} x_e \leq |S| - 1, \quad \forall S \subset V, S \neq \emptyset, V \\
& x_e \in \{0, 1\}, \quad \forall e \in E
\end{aligned}$$

To relax the integer constraint, we can replace it by $x_e \geq 0$.

(b) Given a weighted, *directed* graph $G = (V, E)$ and $s, t \in V$, write an ILP formulation for the problem of computing s-t shortest path distance, and describe how to relax it to an LP. Briefly explain why solving the ILP leads to the optimal solution. **Solution:** In the shortest path, the out-degree and in-degree are equal for all vertices except s and t. In particular, s has out-degree 1 and s has in-degree 1. Using that we obtain the following formulation.

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j) \in E} w_{ij} x_{ij} \\
\text{subject to} \quad & \sum_{(i,k) \in E} x_{ik} - \sum_{(k,j) \in E} x_{kj} = \begin{cases} -1, & k = s \\ 1, & k = t \\ 0, & k \in V \setminus \{s, t\} \end{cases} \\
& x_{ij} \in \{0, 1\}, \quad (i, j) \in A
\end{aligned}$$

We can relax the last constraint to $x_{ij} \geq 0$

(c) Given a weighted, undirected graph and a set of vertices $B = \{v_i\}_{i=1}^k$, the *spider connection* problem asks one to select a minimum weight subgraph that gets all vertices in B connected.

(i) Show how the spider connection problem captures MST and s-t shortest path as its special cases.

(ii) Show that the optimal solution to the spider connection problem is always acyclic, and the edges selected form a single connected component.

(iii) Write an ILP formulation for the spider connection problem, and briefly describe how to relax it to an LP.

Solution: (i). MST is spider connection $B = V$ and s-t shortest path is $B = \{s, t\}$.

(ii). Suppose it has a cycle. Then we can remove any edge to strictly reduce the cost, violating the assumption that the solution is optimal.

Suppose for a contradiction that the edges we selected contains at least 2 connected components. First, if one of the components contain no vertex from B, then we can remove all edges from the solution, reducing its cost; this contradicts optimality. Otherwise, we know that all components contain some vertices from B. However, these components are not connected, and this means that B is not connected by the solution. Hence, the solution is not even feasible, a contradiction.

(iii). Let $\mathcal{S}$ be the set of cuts that contain some vertices in B but not all. So $\mathcal{S}$ is the set of cuts that separate B. Thus, we must select at least one edge for each cut in $\mathcal{S}$. Hence, we modify the MST LP and get

$$\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} w_e x_e \\
\text{subject to} \quad & \sum_{e \in E, e \in \delta(S)} x_e \geq 1 \quad \forall S \in \mathcal{S} \\
& x_e \in \{0,1\} \qquad\qquad \forall e \in E
\end{aligned}$$

We can relax the last constraint to $x_e \geq 0$. It is ok if we also put $x_e \leq 1$ as a constraint, but it is not necessary. (Why?)

(d) Consider the weighted set cover problem. We are given a set of elements U and a collection $\mathcal{T} = \{S_i\}$ of subsets of U, along with weights w_S on the subsets. The goal is to selection a minimum weight collection of sets from $\mathcal{T}$ such that every element in U is in a set (*i.e.*, covered).

Write an ILP formulation for the weighted set cover problem, and briefly describe how to relax it to an LP. **Solution:** Let $\{S_i\}_{i=1}^n$ be the collection of sets.

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} w_i x_i \\
\text{subject to} \quad & \sum_{i : v \in S_i} x_i \geq 1 \quad \forall v \in U \\
& x_i \in \{0,1\} \quad \forall i \in \{1, \dots, n\}
\end{aligned}$$

As usual, we can relax the binary constraint to $x_i \geq 0$. It is ok if we also put $x_i \leq 1$ as a constraint, but it is not necessary. (Why?)

4 Integrality gap

In the last question, we formulated many problems as ILP and then relax it to LP. The requirement is that the ILP must directly lead to an optimal, integral solution. In this question, we investigate a curious phenomenon that the optimal solution of the LP relaxation may be fractional, and the optimal objective can be better than the ILP solution. This is known as the integrality gap of LP. In particular, define the integrality gap of an LP relaxation is

$$\mathsf{IG} = \frac{\mathsf{OPT}(ILP)}{\mathsf{OPT}(LP)},$$

where $\mathsf{OPT}(LP)$ and $\mathsf{OPT}(ILP)$ denote the optimal objective value of the LP and its corresponding ILP.

(a) Given an unweighted, undirected graph $G = (V, E)$, the vertex cover problem asks one to select a minimum set $S \subseteq V$ of vertices such that for each edge, at least one of its endpoints is in S.

(i) Show that the vertex cover problem is a special case of the (unweighted) set cover problem. You need the specify what the universe U and collection of sets S are in the set cover problem.

(ii) Write an ILP formulation for vertex cover, and then describe its LP relaxation.

(iii) Describe a graph where the integrality gap of the above ILP and LP is strictly greater than 1. Compute $\mathsf{OPT}(ILP)$ and $\mathsf{OPT}(LP)$ on this graph.
Hint: What can you do with just 3 *vertices?*

(iv) For each n, describe an n-vertex graph such that the integrality gap approaches 2 as $n \to \infty$.
Hint: Can you generalize your construction for (c) somehow?

Solution: (i). The universe is $U = E$, and the collection of sets is $S = \{E_v\}_{v \in V}$, where E_v denotes the set of edges v is incident on.

(ii).

$$\begin{array}{lll} \text{minimize:} & \sum_{v\in V} x_v & \\ \text{subject to:} & x_u + x_v \geq 1 & \text{for each edge } \{u, v\} \in E \\ & x_v \in \{0, 1\} & \text{for each vertex } v \in V \end{array}$$

We can relax the binary constraint to $x_v \geq 0$.

(iii). Consider a triangle. The integral solution must pick 2 vertices to be feasible. The LP solution, however, sets $x_v = 1/2$ for each v. This gives a fractional objective of $3/2$. So the gap is $\frac{2}{3/2} = 4/3$.

(iv). Consider K_n, the complete graph on n vertices. The integral solution must pick $n - 1$ vertices; otherwise, if some pair u, v are both not picked, then edge (u, v) are not covered, violating the constraint. The fractional optimal solution, on the other hand, is to set $x_v = 1/2$ for each v. This gives an objective value of $n/2$. Thus, the integrality gap is $\frac{n-1}{n/2} = 2(1 - /1/n)$, which approaches 2 for large n.

(b) Given an unweighted, undirected graph $G = (V, E)$, the maximum independent set problem asks for a maximum set of vertices such that no pair is connected by an edge.

(i) Write an ILP formulation for maximum independent set, and then describe its LP relaxation.

(ii) For each n, describe an n-vertex graph such that the integrality gap between the ILP and LP is $\Theta(1/n)$. (This means that the LP give a much larger objective value than ILP.) Compute $\mathsf{OPT}(ILP)$ and $\mathsf{OPT}(LP)$ on this graph.

Solution: (i). The ILP is given by

$$\begin{array}{lll} \text{maximize:} & \sum_{v\in V} x_v & \\ \text{subject to:} & x_u + x_v \leq 1 & \text{for each edge } \{u, v\} \in E \\ & x_v \in \{0, 1\} & \text{for each vertex } v \in V \end{array}$$

Again, we can relex the last constraint as $x_v \geq 0$.

(ii). Again, consider the complete graph of n vertices. The integral optimal solution can only choose 1 vertex. For the LP, it can produce a fractional solution that sets $x_v = 1/2$ for each v, so the objective is $n/2$. This gives a gap of $2/n$.

5 Duality

In this problem, we explore linear programming duality.

(i) Consider the LP relaxation for the unweighted vertex cover problem. Write its dual LP.

(ii) Interpret the variables and constraints of the dual LP. What do they correspond to on the graph? (You don't have to be formal. Just explain the idea.)

(iii) Suppose we have an integral feasible solution to the primal LP, with objective value $P > 0$, and an arbitrary feasible solution to the dual LP, with objective value $D > 0$. Further, assume $P/D \leq c$. Let $OPT > 0$ denote the objective value of the optimal integral solution to the primal. Show that $P/OPT \leq c$, that is, the primal solution is approximately optimal by a factor of c. (This statement holds in general for any primal/dual LP pair, not only in the context of vertex cover.)
Hint: Use weak duality

(iv) *Fun and pretty hard challenge*: Use the idea above to approximate the (integral) vertex cover problem by a factor of 2.

(v) Consider the LP relaxation for weighted set cover problem that you wrote earlier. Write its dual LP.

Solution:

(i) Let E_v all the edges incident on vertex v. The dual LP is given by

$$\begin{aligned} \text{maximize} \quad & \sum_{e \in E} y_e \\ \text{subject to} \quad & \sum_{e \in E_v} y_e \leq 1 \quad \forall v \in V \\ & y_e \geq 0 \qquad \forall e \in E \end{aligned}$$

(ii) Fractional maximum matching on graph.

(iii) By weak duality, the objective of any primal solution is at least the cost of any dual solution. This means $P \geq D$ and $OPT \geq D$. Furthermore, since P is the objective of an integral solution and OPT is the *optimal* objective among all integral solutions, $P \geq OPT$. Hence $P \geq OPT \geq D$. This implies that $P/OPT \leq P/D \leq c$.

(iv) The algorithm is to initialize all $y_e, x_v = 0$. Then, we choose an arbitrary dual variable y_e. We increase y_e continuously until one of the related dual constraints becomes tight. Suppose the primal variable corresponding to this tight dual constraint is x_v, we set x_v to 1. Repeat this process until all primal constraints are satisfied. Note that the algorithm constructs a feasible dual solution and an *integral*, feasible primal solution. To show the approximation factor, we bound the ratio between the primal and dual objectives:

$$\begin{aligned} P &= \sum_{v \in V : x_v = 1} x_v \\ &= \sum_{v \in V : x_v = 1} \sum_{e \in E_v} y_e \\ &\le 2 \sum_{e \in E} y_e \\ &= 2D \end{aligned}$$

The second equality is because that a primal variable is set to 1 iff its corresponding dual constraint is tight. The inequality follows since each edge contains only 2 vertices, which means each dual variable at most appears twice in the entire double sum.

(v) Let $\mathcal{U}$ be the universe and $\mathcal{T}$ be the collection of subsets of $\mathcal{U}$. The dual of weighted set cover is given by

$$\begin{aligned} \text{maximize } & \sum_{e \in \mathcal{U}} y_e \\ \text{subject to } & \sum_{e \in S} y_e \le w_S \quad \forall S \in \mathcal{T} \\ & y_e \ge 0 \qquad \forall e \in \mathcal{U} \end{aligned}$$