DeJuan Anderson May 9, 2014
6.046 Problem 5-6
Collaborators: *Stephanie New, Kang Zhang* Recitation: Kang Zhang, 12:00

1 Devise an efficient algorithm that given $C_i, \ldots, n$ and V finds a feasible set of contracts that maximizes the profit. The runtime of your solution should be polynomial in n and P.

This is a version of the Knapsack problem, but requiring a runtime polynomial in different terms; instead of requiring a runtime polynomial in the number of items and the space of the bag, we need it in the number of items and the value of each bag.

As such, this is a dynamic programming problem.

Let us define a function, call it V'(i,p), and let this function indicate the minimum volume required for profit p.

Now we must come up with a recursion that allows us to break the problem into smaller subproblems, solve those, and use the solutions to solve larger and larger subproblems until we arrive at the solution for V'(i,p).

Our choices for each ghost are as follows:

1) We don't capture that ghost at all. 2) We capture that ghost, acquiring its profit and consuming the volume needed to capture it.

We will model these two cases in our recurrence. We will use a top-down recursion:

$$V'(i+1,p) = min(V'(i,p), A(i,[p-p_{i+1}]+v_{i+1}) \quad (1)$$

The first term is the outcome where we do capture the ghost from the previous step.

The second term is the outcome where we don't capture it; we lose the profit it would have gained us, but gain the volume that ghost would have consumed.

If we recurse on this equation, solving all the subproblems, we will be able to minimize the space consumed for profit p. Our runtime for this is O(nP): N for the number of contracts, P for the profit we are searching for.

2 Design an algorithm that, given a precision parameter ϵ ¿ 0, the set of contracts C_i, and the total available metaphysical volume V, finds a set of contracts within 1-ϵ of the most profitable set. Runtime should be polynomial in n and 1/$epsilon$

We will break this up into a sequence of equations and use the relations between them to prove this.

First, we must create a scaling factor for the profit p, so that we can run our previous algorithm with the scaling factor in place.

For any given profit p_i, let our scaled version, p_i' be the following:

$$p_i' = \left\lfloor p_i \frac{n}{\epsilon p} \right\rfloor \tag{2}$$

To aid in our explanation, let us define the following variables:

n = the total number of contacts

ϵ = scaling factor

P = maximum profit obtainable from one ghost

S = optimal set of ghosts

S' = set of ghosts chosen by our approximation

We need to bound the result from this scaling. Notice that the scaling equation for p_i' is floored. If we remove the floor, we obtain an upper bound, since if the result is fractional, the floor removes it:

$$\sum_{i \in S'} P_i' \leq \sum_{i \in S} p_i \frac{n}{\epsilon p} \tag{3}$$

Further, note that if we subtract 1 from $p_i * \frac{n}{\epsilon p}$, we get a term that is at most equal to p_i':

$$\sum_{i \in S} p_i \frac{n}{\epsilon p} - 1 \leq \sum_{i \in S'} P'_i \tag{4}$$

Combining these statements, we obtain:

$$\sum_{i \in S} p_i \frac{n}{\epsilon p} - 1 \leq \sum_{i \in S'} P'_i \leq \sum_{i \in S} p_i \frac{n}{\epsilon p} \tag{5}$$

Now note that $\sum_{i \in S} p_i$ is precisely the optimal solution, and $\sum_{i \in S'} p'_i$ is our approximation of that solution. Further, $\sum_{i \in S} -1$ = - $|S|$, where $|S|$ is the size of the set S.

Let ALG represent our approximation, and let OPT represent the optimal solution. If we sub these three equivalences into Equation (5), we obtain:

$$\frac{n}{\epsilon p} OPT - |S| \leq ALG \leq OPT \frac{n}{\epsilon p} \tag{6}$$

Now, multiply through by $\frac{\epsilon p}{n}$ to isolate OPT on the right side:

$$OPT - \frac{\epsilon p}{n} |S| \leq \frac{\epsilon p}{n} ALG \leq OPT \tag{7}$$

Note that by the construction of the problem, $|S|$ must necessarily be less than n, so the quantity $\frac{|S|}{n}$ is less than one.

We can then sub this in to find a new equation, since ϵp is greater than $\epsilon p * \frac{|S|}{n}$:

$$OPT - \epsilon p \leq OPT - \frac{\epsilon p}{n} |S| \leq \frac{\epsilon p}{n} ALG \leq OPT \tag{8}$$

Removing the second term in the above inequality:

$$OPT - \epsilon p \leq \frac{\epsilon p}{n} ALG \leq OPT \tag{9}$$

Finally, note that p $\leq$ OPT. Since p is the maximum profit obtainable from one ghost, if the optimal decision is taking p and only p then p = OPT. If we have room for more ghosts after including p, p is less than OPT.

Similarly, if P is not in OPT, then OPT contains some sequence of ghosts such that the total profit amassed by capturing those is in excess of P, meaning once more p is less than OPT.

By plugging in the fact that $p \leq OPT$, we finally obtain the desired bound, proving that this approximation will be within the factor required:

$$\begin{aligned} OPT - \epsilon OPT &\leq \frac{\epsilon p}{n} ALG \leq OPT \\ OPT(1-\epsilon) &\leq \frac{\epsilon p}{n} ALG \leq OPT \end{aligned} \tag{10}$$

To carry out this approximation, we replace all p_i with p'_i, and run the algorithm described in section 1. After we get the result from this approximation, we can scale it back to the original profit numbers used by simply multiplying by $\frac{n}{\epsilon p}$. Q.E.D.