Approximation Algorithms Lecture 11

Contents

1 Recap

Load balancing PTAS. Read the book to get the correct proof.

2 Content

2.1 Fully polynomial time approximation scheme (FPTAS) for knapsack problem

There is a knapsack. You're given n objects, the i^{th} object has weight w_i and value p_i, both in $\mathbb{R}^+$.

The knapsack can carry a weight of W. Pick a subset of objects of weight at most w and having maximum value.

Suppose the weights are real numbers and the values are positive integers.

Dynamic programming: $dp(i, j)$ is the minimum weight of a subset of $\{1, \ldots, i\}$ which has a profit of j.

Then we have $dp(i, j) = \min\{dp(i-1, j), w_i + dp(i-1, j-p_i)\}$. The time required is $O(nP)$, which is not polynomial time in the input, since the size of input is $O(n \log W_{max} + n \log P_{max})$, and P is the total profit, which is at least P_{max}.

Now our aim is to get a polynomial time approximation. So we will build a $(1+\varepsilon)$ approximation that has running time polynomial in the input length and $\frac{1}{\varepsilon}$.

Idea: Scale profits by k, which will be determined later. Then $p'_i = \lfloor \frac{p_i}{k} \rfloor$. So with the scaled profits, the running time becomes $O(nP/k)$.

How about the profit of the solution obtained by this method? How does it compare to **OPT**?

Let I' be the set of objects picked by this algorithm, with profit function p', and let I be the set of objects picked in the optimal solution, with profit function p.

We have $profit(I') = \sum_{i \in I'} p_i \geq k \sum_{i \in I'} p'_i \geq k \sum_{i \in I} p'_i > k \sum_{i \in I} \frac{p_i}{k} - 1 = \sum_{i \in I} p_i - k|I| = \mathbf{OPT} - k|I| \geq \mathbf{OPT} - kn$.

So we want $kn < \varepsilon \mathbf{OPT}$, i.e., $k \leq \frac{\varepsilon \mathbf{OPT}}{n}$.

Set $k = \frac{\varepsilon P}{n}$. Note that $P \leq \mathbf{OPT}$.

This inequality is probably reversed

The running time is polynomial in $n, 1/\varepsilon$. These are called fully polynomial time approximation schemes.

Theorem 1

There can be no PTAS for bin packing.

Proof. Given a bin-packing instance, it is NP-hard to determine if the objects can be packed in two bins. This is by showing a reduction to the subset sum problem.

Now suppose we have a PTAS for bin packing. Then we claim that we can solve this problem.

Consider the PTAS for bin packing. Find an approximation of $\varepsilon < 1/2$ using the PTAS. If the PTAS for bin packing has 2, then yes, else no. This solves the subset sum problem. $\square$

We will next look at asymptotic PTAS.