Approximation Algorithms Lecture 9

Contents

1 Recap

k-center problem.

2 Content

2.1 Load balancing revisited

Let's come back to load balancing. PTAS = polynomial time approximation scheme. We will try to do some sort of packing.

2.2 Bin packing

Given n objects of sizes $0 \leq s_1, \ldots, s_n \leq 1$. What is the minimum number of bins of size 1 needed to pack the objects.

Bin packing can be solved optimally in $O(n^{2s})$ time, where s is the number of distinct sizes.

Proof. Let n_1 be the number of objects of size s_1. An instance of bin packing is a vector $(n_1, n_2, \cdots, n_s)$ and the sizes $s_1, \cdots, s_s$.

Let $B(o_1, \cdots, o_s)$ be the number of bins needed to pack o_i objects of size s_i. We are interested in $B(n_1, \cdots, n_s)$.

$$B(o_1, o_2, \cdots, o_s) = \min_{c_1, \cdots, c_s \geq 0, \sum_{i=1}^{s} c_i s_i \leq 1} B(o_1 - c_1, \cdots, o_s - c_s) + 1$$

with $B(0, 0, \cdots, 0) = 0$.

Let B be an $s-$dimensional matrix whose $(o_1, \cdots, o_s)$-th entry is $B(o_1, \cdots, o_s)$. This has $O(n^s)$ entries. To compute an entry, we would have to check all possible choices of $(c_1, \cdots, c_s)$ where $c_i \leq o_i$. Total number of choices is $\prod_{i=1}^{s}(1 + o_i) \leq (n+1)^s$. Total time required to fill this table is thus $O(n^{2s})$. □

Is there a way of assigning jobs to machines such that the makespan is at most T? TO check this, we do $s_i = p_i/T$, and if the number of bins is $\leq m$, then the answer is yes, else it is no. Now do a binary search on T to solve for the final answer.

This gives an $O(n^{2s} \cdot T(\text{binary search}))$ time algorithm for load balancing where s is the number of different processing times.

Rounding processing times

1. Ignore all jobs with processign times less than ϵT.
2. Round down processing times to the nearest value to $\varepsilon T(1+\varepsilon)^k$.

Note that $\varepsilon(1+\varepsilon)^{k-1} = 1$, so $k = \lceil \log_{1+\varepsilon} \frac{1}{\varepsilon} \rceil$.

Given a bin-packing solution that uses $\leq m$ bins, we can obtain a solution for load balancing that has makespan at most $T(1+\varepsilon)$. For the jobs that we didn't ignore, we get this trivially, since the weight just after the weight we rounded it down to was more than the original size. So we get a makespan of $T(1+\varepsilon)$. It might be possible that the small jobs can possibly exceed the whole thing. We'll do it in the next class.